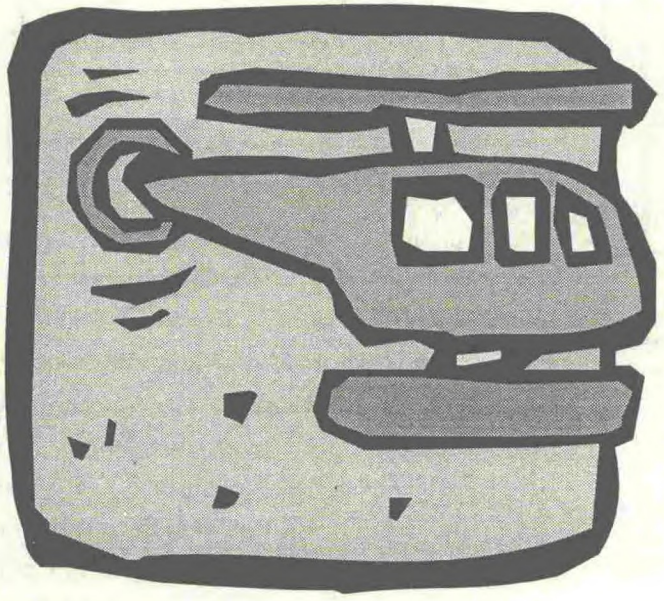
ดร.ปัญญา เปรมปรีดิ์

โปรแกรมบอกเวลาทั่วโลก

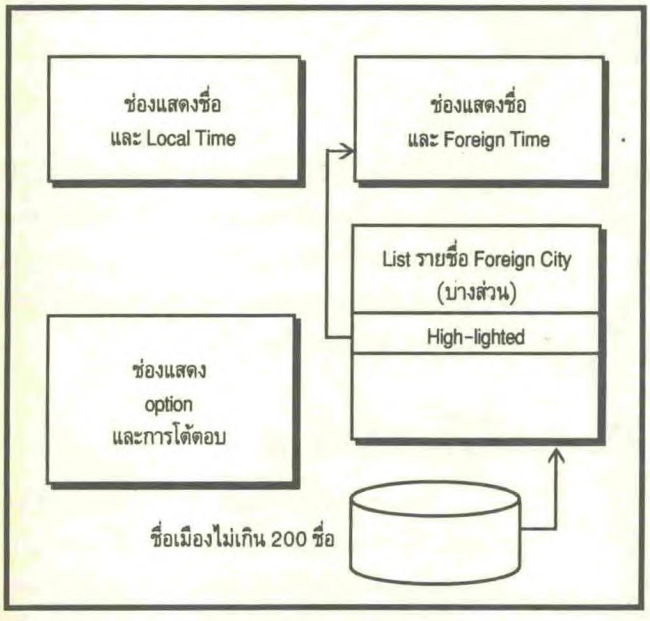


นำเสนอโจทย์และโปรแกรมตัวอย่าง ที่เกี่ยวกับการบอกเวลา

ไม่ได้เขียนโปรแกรมมาให้สมาชิกอ่านเล่นเสียหลายเดือน หวังว่าคงจะยังไม่ลืมกัน แต่พอหยิบเอาหนังสือเรื่อง "โจทย์เกมโอลิมปิกคอมพิวเตอร์" ของอาจารย์ยืน (ยืน ภู่วรวรรณ) มาอ่านดูแล้วก็เกือบหมดแรง ทั้งนี้เพราะโจทย์เหล่านั้นดูมันยากเย็นกว่าการพัฒนาระบบให้ยูสเซอร์ใช้งานไปอีกหลายเท่าตัว ผมดูแล้วต้องขอยกย่องนักเรียนไทยที่อุตส่าห์เขียนแข่งขันจนได้เหรียญมาหลายเหรียญ โจทย์พวกนั้นมันยากจริง ๆ เมื่อเจอกับปัญหาแบบนั้นผมก็เลยต้องคิดตั้งโจทย์ขึ้นมาเอง โจทย์ของผมอันนี้ถ้ามีเวลาพัฒนาให้ดี ๆ ก็อาจขายได้ โจทย์นี้เกิดขึ้นกับผู้ที่ต้องเดินทางไปต่างประเทศโดยเครื่องบิน กล่าวคือเวลาเราขึ้นเครื่องบินจากเมืองหนึ่งไปยังอีกเมืองหนึ่งนั้นเราจะไม่รู้ว่าเครื่องมันบินอยู่กี่ชั่วโมงกันแน่ ในตั๋วเครื่องบินจะแสดงเวลาขึ้นเครื่องกับเวลาถึงเป้าหมายเป็น "Local Time" ของเมืองทั้งสอง แต่เขาไม่ได้บอกว่าเมืองสองเมืองนั้นใช้เวลาตรงกันหรือไม่ การเอาเวลาขึ้นเครื่องกับลงเครื่องมาลบกันเฉย ๆ จะไม่ใช่เวลาที่ต้องนั่งบนเครื่องบิน ยกตัวอย่างเช่น เที่ยวบินที่ TG910 ออกจากกรุงเทพฯ 00:30 น. จะถึงลอนดอนตอน 06:20 น. ถ้าเราเอาตัวเลขสองตัวนี้มาลบกันเราก็อาจคิดว่าใช้เวลาแค่ 5 ชั่วโมง 50 นาที แต่ในความเป็นจริงนั้นเราต้องนั่งอุดอู้อยู่บนเครื่องกว่าครึ่งวัน ในขณะเดียวกันที่เที่ยวบินที่ TG620 จากกรุงเทพฯ 11:10 น. จะถึงโอซาก้าประเทศญี่ปุ่นตอน 21:00 น. ซึ่งอาจทำให้เราคิดว่าต้องนั่งอยู่ 10 ชั่วโมง แต่แท้จริงแล้วเราบินสวนทางกับพระอาทิตย์ เราใช้เวลา 3-4 ชั่วโมงเท่านั้น

ตั้งโจทย์ให้คิด

โจทย์ของผมก็คือ ผมต้องการได้โปรแกรมที่บอกเวลาของเมืองที่เราอยู่กินในขณะนี้ (หรือเมืองที่เรากำลังจะออกเดินทาง) กับเมืองอื่นที่เราสนใจ (หรือคือเมืองที่เราจะเดินทางไปถึง) เมืองที่เราอยู่กินนั้นเราเรียกว่า "Local City" ส่วนเมืองอื่นที่เราสนใจนั้นเรียกว่า "Foreign City" เจ้าเวลาของโลคัลซิตี้นี้เราหาได้โดยติดต่อกับโอเปอเรติงซิสเท็ม (DOS เป็นต้น) ส่วนเวลาของฟอรีนซิตี้นั้นเราสามารถคำนวณได้โดยเรียกข้อมูลจากจานแม่เหล็ก วิธีการคำนวณนั้นจะเป็นอย่างไรเดี๋ยวผมจะเล่าให้ฟัง ผมอยากจะอธิบายความต้องการอย่างย่อ ๆ เสียก่อน โจทย์ต้องการให้เอาเวลาของโลคัลซิตี้ขึ้นมาแสดงบนจอภาพด้านซ้ายมือและเอาเวลาของฟอรีนซิตี้ขึ้นมาโชว์บนจอด้านขวามือ ใต้ของเวลาของฟอรีนซิตี้จะเป็นรายชื่อของเมืองต่าง ๆ จำนวน 10 หรือ 11 ชื่อ และชื่อที่ตรงกับฟอรีนซิตี้ที่กำลังโชว์อยู่บนจอนั้นจะต้องแสดงด้วยสีเด่น (High-lighted) จากภาพดังกล่าวนี้ให้เราเลือกเปลี่ยนตำแหน่งที่แสดงสีเด่นด้วยการกดปุ่ม U หรือ D (U= Curssor Up, D= Curssor Down) เมื่อเราเลื่อนตัวชี้ตำแหน่งของฟอรีนซิตี้ลงไปจนถึงชื่อล่างสุดแล้วและเรายังมีต่อไปอีกในจานแม่เหล็กก็ให้เลื่อนรายชื่อขึ้นไป (Scroll Up) ในทำนองเดียวกัน เมื่อเราเลื่อนตัวชี้ (High-light หรือ curssor) ขึ้นไปถึงบนสุดของบนหน้าจอแล้วก็ให้เลื่อนรายชื่อลง (Scroll Down) การเลื่อนรายชื่อขึ้นและลงนี้เมื่อไม่มีชื่อในจานแม่เหล็กให้เลื่อนได้แล้วให้หยุดอยู่แค่นั้น และให้ทำเสียง "BEEP, BEEP" เพื่อเตือนว่าขึ้นต่อไปหรือลงต่อไปไม่ได้แล้ว สำหรับรายชื่อเมืองที่จะเก็บไว้ในจานแม่เหล็กนั้นให้เตรียมไว้ไม่เกิน 200 ชื่อ ที่จำกัดไว้เพื่อแสดงว่ามันเป็นเพียงโปรแกรมตัวอย่าง สำหรับงานในชีวิตจริงอาจต้องเพิ่มเป็น 2,000-3,000 ชื่อ จึงจะครอบคลุมได้ทั่วโลกจริง ๆ



รูปที่ 1 แสดงการจัดวางข้อมูลต่าง ๆ บนจอภาพ

ภาพบนหน้าจอจะเป็นดังแสดงในรูปที่ 1 ท่านผู้อ่านควรระลึกอยู่เสมอว่า **นี่เป็นแค่โปรแกรมตัวอย่าง** หน้าตามันจึงไม่สะสวยเท่าใดนัก เราควรจะดูในด้านหลักการและโครงสร้างของโปรแกรมเสียก่อน เมื่อโครงสร้างเสร็จแล้วเราจะปรับปรุงให้สวยแค่ไหนก็ย่อมได้

เพิ่มความมันส์

ดูโจทย์แล้วท่านผู้อ่านอาจบอกว่า "หมูมาก" ด้วยเหตุนี้ผมจึงใคร่ขอทำให้มันยากยิ่งขึ้นสัก 2-3 อย่าง ประการแรกก็คือ เวลาที่แสดงในช่องแสดง "Local Time" และ "Foreign Time" นั้นให้แสดง "วัน เดือน ปี วันของสัปดาห์ (อาทิตย์, จันทร์,...) ชั่วโมง นาที และวินาที" ให้ครบถ้วนทั้งสองเมือง และให้เปลี่ยนไปตามนาฬิกาที่เปลี่ยนไป หรือคือ ตัวเลขบอกวินาทีจะเปลี่ยนไปทุก ๆ วินาที และเมื่อเปลี่ยนจาก 59 ไปสู่ 00 ก็จะต้องขยับเจ้าตัวเลขนาที... ฯลฯ ตามไปด้วย

สำหรับความต้องการที่จะเพิ่มความยากขึ้นไปเป็นประการที่สองก็คือ โปรแกรมจะต้องสามารถให้เราปรับปรุงและแก้ไขแฟ้มข้อมูลที่เก็บรายชื่อเมืองได้ด้วย การปรับปรุงแก้ไขทำได้ 3 อย่าง คือ

1. เพิ่มชื่อเมืองเข้าไป (=ADD)
2. ลบชื่อเมืองออกทิ้ง (=DELETE)
3. เปลี่ยนเมืองที่เป็น "Local City" (= SET LOCAL CITY)

เพื่อให้มันยุ่งยากและสับสนยิ่งขึ้น ผมก็จะบังคับด้วยว่าในรายชื่อของเมืองต่าง ๆ นั้นเราจะมีอยู่เมืองหนึ่งที่ให้ชื่อว่า "G.M.T." นี่คือคำย่อของคำว่า Greenich Mean Time และชื่อนี้จะลบทิ้งไม่ได้

สำหรับข้อกำหนดข้อสุดท้ายก็คือ รายชื่อเมืองที่เก็บไว้ในจานแม่เหล็กนั้นจะต้องจัดเรียงตามตัวอักษรอยู่ตลอดเวลา เวลาเราเพิ่มชื่อเมืองเข้ามาก็จะต้องจัดแทรกลงให้ถูกที่ (โดยการเลื่อนชื่อที่มีตัวอักษรสูงกว่าลงไป 1 ตำแหน่ง) และเวลาเราจะสั่งลบทิ้งก็จะต้องเลื่อนชื่อที่อยู่ด้านล่างขึ้นมาแทนที่ให้เต็มโดยไม่มีช่องว่าง

มาตีความกันดู

ถึงตอนนี้ผมก็อยากจะให้พวกเราลองมาดูว่าข้อกำหนดต่าง ๆ มันจะเพิ่มความยุ่งยากได้อย่างไร และนี่ก็ถือเป็นการซักซ้อมความเข้าใจของเราไปในตัว

เรามาดูเรื่องของการกำหนดให้แสดงวัน เดือน ปี ของเมืองทั้งสองเสียก่อน ปัญหาก็คือ ลูกโลกของเรานี้มันเป็นพื้นที่ที่วนรอบมาบรรจบกันเป็นวงกลม ดังนั้นวันและเวลาของเมืองต่าง ๆ จะไม่ตรงกัน สิ่งที่อาจเป็นได้ก็คือเมืองโลคัลซิตี้กับเมืองฟอรีนซิตี้อาจเป็นคนละวันก็ได้ มันเป็นไปได้แม้กระทั่งในกรณีที่สองเมืองนั้นอยู่ห่างกันเพียงไม่กี่ร้อยกิโลเมตร ยกตัวอย่างเช่น ขณะที่กรุงเทพฯเราเป็นเวลาที่เพิ่งผ่านเที่ยงคืนมาเล็กน้อย แต่เมืองย่างกุ้งของเหมียนหม่าจะยังเป็นวันก่อนอยู่ และเรื่องนี้ถ้าเกิดไปตรงกับวันขึ้นปีใหม่ของไทยเราพอดี (1 มกราคม 1997, 00:00:01 น. เป็นต้น) ความแตกต่างก็อาจเป็นทั้งเรื่องของวัน เดือน และปี เรื่องนี้จะยุ่งยากขึ้นไปอีกเมื่อเราขึ้นเดือนใหม่ไปแล้ว แต่ย่างกุ้งและเมืองอื่น ๆ อีกหลายเมืองจะเป็นเดือนเก่าอยู่ เราจึงต้องคำนึงถึงเรื่องจำนวนวันในแต่ละเดือนเข้าไปด้วย โลจิกเพื่อคิดและคำนวณวัน เดือน และปีของฟอรีนซิตี้นี้จะยาว 2 ถึง 3 หน้ากระดาษได้

ปัญหาความยุ่งยากประการที่สองก็คือ การบังคับให้แสดงเวลาที่เปลี่ยนไปตามเวลาจริง เรื่องนี้ผมเคยเขียนมาให้อ่านกันครั้งหนึ่งแล้ว แต่หลาย ๆ ท่านอาจลืมไปแล้วก็ได้ วิธีการในเรื่องนี้ก็คือ เราจะต้องวนอยู่ในลูปอันหนึ่ง ภายในลูปอันนี้เราจะเช็คว่ามีการกดแป้นพิมพ์บ้างหรือไม่ ซึ่งเราทำได้โดยใช้คำสั่งว่า IF KEYPRESSED THEN.... เจ้าตัวแปรที่ชื่อว่า KEYPRESSED นั้นเป็นตัวแปรแบบบูลีน (Boolean Variable) ซึ่งตัวโอเปอเรติงซิสเต็มจะเป็นตัวเซตให้เรา ถ้ามีการกดแป้นเราก็จะให้โปรแกรมกระโดดไปจัดการกับรหัสแป้นที่ผู้ใช้กดเข้ามา แต่ถ้าไม่มีการกดแป้นพิมพ์เราก็จะให้โปรแกรมมันเดินต่อโดยทำการติดต่อขอเวลาและวันเดือนปีจากโอเอส แล้วก็นำมาคำนวณและแสดงผลบนจอภาพ เราจะแสดงผลก็เมื่อค่าของเวลามันเปลี่ยนไป นี่จะช่วยลดภาระการแสดงผลบนจอและจะช่วยให้โปรแกรมมันพร้อมที่จะรับคำสั่งจากผู้ใช้ทางแป้นพิมพ์อยู่เสมอ

มาดูในเรื่องของการอัพเดตแฟ้มข้อมูลกันบ้าง แฟ้มข้อมูลนี้จะต้องมีไว้ 200 เรคอร์ด แต่ละเรคอร์ดจะเตรียมเนื้อที่เอาไว้สำหรับเก็บชื่อและรหัสบางอย่าง รหัสอันแรกที่ต้องสร้างไว้ก็คือ การบอกว่าเรคอร์ดนี้เป็นเรคอร์ดว่างเปล่า หรือเป็นเรคอร์ดที่เป็นดาต้าของเมืองตามปกติ หรือเป็นดาต้าของเมืองที่เป็นโลคัลซิตี้ ปัญหาของการอัพเดตก็คือ ก่อนที่เรา

จะสั่งลบข้อมูลทิ้ง (DELETE) นั้นเราต้องเช็ครหัสตัวนี้เสียก่อน ถ้ามันเป็นโลคัลซิตี้อยู่พอดีเราจะสั่งลบทิ้งไม่ได้ ถ้าเราลบออกไปก็จะทำให้ไม่มีโลคัลซิตี้สำหรับนำไปแสดงบนจอภาพ ในขณะเดียวกัน ถ้าชื่อของมันเป็น "G.M.T." เราก็ลบทิ้งไม่ได้ เรื่องนี้จึงทำให้โลจิกของเราขาวยิ่งขึ้น แต่หลังจากที่เราเช็คแล้วผ่านมาโดยตลอด คือเป็นประเภทชื่อเมืองธรรมดา เราก็ลบออกได้ แต่ก็จะมาหนักใจในเรื่องการเลื่อนชื่อเมืองอื่น ๆ ขึ้นมาแทนที่ การเลื่อนชื่อเมืองนี้มันหมายถึงว่าจะต้องมีการอ่านข้อมูลจากจานแม่เหล็ก แล้วก็บันทึกกลับลงไปโดยเลื่อนตำแหน่งที่ตั้งไปเล็กน้อย แล้วก็จะต้องวนให้ครบจนถึงเรคอร์ดสุดท้าย การทำงานที่มีทั้งอ่าน (READ) และบันทึก (WRITE) สลับกันนี้ใช้เวลา มันเกิดความเสี่ยงในเรื่องไฟฟ้าดับ ถ้าไฟมาดับเอาในช่วงนี้ก็จะทำให้ไฟล์ของเราเละตุ้มเปะไปได้ เรื่องนี้จะแก้ไขอย่างไรเดี๋ยวผมจะเฉลยให้ฟัง

ขอกลับมาดูปัญหาอีกรูปแบบหนึ่ง ปัญหาคือการแสดงรายชื่อเมืองอยู่บนจอภาพ นี่เป็น "list" ชนิดหนึ่ง แต่เป็น "short list" หรือคือเป็นเพียงบางส่วนของรายชื่อทั้งหมด จอภาพของเรานั้นมีเนื้อที่จำกัด แต่ผู้ใช้โปรแกรมก็อยากจะให้เห็นรายชื่อของเมืองสักจำนวนหนึ่ง ทั้งนี้เพราะเขาจะเลือกได้ว่าจะเอาเมืองไหนมาเป็นฟอรีนซิตี้ และรายชื่อเหล่านี้จะต้องจัดเรียงตามตัวอักษรด้วย ทั้งนี้เพื่อว่าผู้ใช้จะได้รู้ว่าควรเลื่อนเคอร์เซอร์ไปทางไหน ยกตัวอย่างเช่น ในขณะนี้เรามี LONDON เป็นฟอรีนซิตี้ และเราต้องการเปลี่ยนไปเป็น AUCKLAND อย่างนี้ก็จะต้องเลื่อนเคอร์เซอร์ขึ้นข้างบน ทั้งนี้เพราะอักษร A ของ AUCKLAND มันอยู่ก่อนอักษร L ของ LONDON ปัญหาของการแสดง "short list" นี้เป็นปัญหาพื้นฐาน เราต้องใช้ในงานหลายอย่าง การทำงานของโปรแกรมนั้นมีหลายสถานะ เช่น ในขณะที่เคอร์เซอร์อยู่กลาง ๆ ของ "short list" เราจะเลื่อนแค่ตัวเคอร์เซอร์ (ในโจทย์นี้เราใช้การทำ High-light แทนตัวเคอร์เซอร์) ขึ้นหรือลง แต่พอเคอร์เซอร์ขึ้นไปติดขอบบนหรือลงไปติดอยู่ขอบล่างของ "short list" เราก็จะทำงานไปอีกแบบหนึ่ง คือปล่อยตำแหน่งเคอร์เซอร์ไว้ที่เก่า แต่เลื่อนตัว "short list" ลงมาหรือขึ้นไป การเลื่อนรายชื่อจะตรงกันข้ามกับการเลื่อนตำแหน่งของเคอร์เซอร์เสมอ คนที่จะเขียนโปรแกรมชนิดนี้ต้องมีโลจิกที่ดี มีความแม่นยำในเชิงความคิด มันจึงเป็นโจทย์ที่ดีอย่างหนึ่ง

ข้อกำหนดที่ต้องซักแซกถาม

อ่านมาถึงตรงนี้บางท่านอาจบอกว่า "โอเค, ผมพร้อมที่จะเขียนโปรแกรมแล้ว" แต่หลาย ๆ ท่านน่าจะ "ยังมีข้อสงสัย" นี่เป็นข้อแตกต่างระหว่างโปรแกรมเมอร์ชั้นดีกับชั้นเลว ใครที่บอกว่าพร้อมแล้วนั้นจะไปเจอกับปัญหาอีกหลายอย่าง ประการแรกก็คือ ในระหว่างที่ยูสเซอร์กำลังขอทำการอัพเดตข้อมูลอยู่นั้นเราต้องรักษาระบบการแสดงเวลาทุก ๆ วินาทีอยู่อีกหรือไม่ เรื่องนี้เราจะพบว่าถ้าโจทย์จะบังคับให้แสดงเวลาอยู่ต่ออีกก็จะวุ่นวายมากมายเลยทีเดียว เช่น เราต้องไปวนลูปใหม่อีกลูปหนึ่ง แต่ในลูปอันนี้การตีความของแป้นที่ถูกกดจะยุ่งยาก เพราะมันจะปะปนกับการอินพุตชื่อเมืองและข้อมูลต่าง ๆ เรื่องนี้โปรแกรมเมอร์ควรถามเอาไว้เสียก่อน ถ้าไม่ถามแล้วไปคิดเอาเองว่าต้องทำ ก็จะเสียเวลามากขึ้น สำหรับโจทย์ตัวอย่างอันนี้ผมขอกำหนดว่า "ในระหว่างที่มีการอัพเดตไฟล์จะไม่ต้องแสดงเวลาตามเวลาจริง" เหตุผลก็คือยูสเซอร์จะไม่มีเวลามาดูตัวเลขบอกเวลาอยู่แล้ว เขากำลังเพ่งอยู่ที่การโต้ตอบกับเครื่องเพื่ออัพเดตข้อมูลอยู่

โล่งไปหน่อยหนึ่ง แต่ก็ยังจะมีคำถามที่ควรถามอีก เช่น เมื่อไฟล์เต็ม (มีข้อมูลครบ 200 เรคอร์ด) แล้วยูสเซอร์ยังอยากจะเพิ่มข้อมูลอีก (=ADD) จะทำอย่างไร คำตอบก็คือ "ด่ากลับออกมาซิครับ"

จริง ๆ แล้วยังจะมีคำถามได้อีกมาก แต่ผมขอตัดมาถึงคำถามสุดท้ายเลย คือ การสั่งลบข้อมูลทิ้ง (DELETE) นั้นยูสเซอร์ต้องคีย์ชื่อเมืองเข้ามาหรือเปล่า เรื่องนี้ถ้าผมตอบว่า "ใช่" ก็จะทำให้ซับซ้อนขึ้นอีกมาก เช่น ถ้าคีย์ไม่ตรงกันจะทำยังไง การตอบว่า "ไม่ใช่" ก็ยุ่งยากเช่นกัน คือผมต้องชี้แจงว่าจะอัพเดตแบบ "DELETE" ก็ให้หมายถึงการลบเมืองที่กำลังเป็นฟอรีนซิตี้ในขณะนั้นเลย วิธีนี้จะไม่ทำให้ผิดตัว แนวคิดอันนี้ใช้กับการเซตให้เมืองใดเป็นโลคัลซิตี้ด้วย นั่นคือ เมื่อยูสเซอร์ต้องการเปลี่ยนเอาเมืองอื่นมาเป็นโลคัลซิตี้ก็ต้องเลื่อนเคอร์เซอร์ไปยังเมืองดังกล่าวเสียก่อน ถึงแล้วให้กดปุ่มเลือกการทำ "File Maintenance" และเลือกทำฟังก์ชัน "Set Local City" เจ้าเมืองนั้นก็จะได้เป็นโลคัลซิตี้ทันที และเนื่องจากการเปลี่ยนแปลงข้อมูลในไฟล์จะกระทบต่อจอภาพได้หลายอย่าง ผมก็ขอบอกไปด้วยว่า "หลังจากทำการอัพเดตไฟล์แล้วให้เคลียร์หน้าจอและเริ่มต้นกันใหม่หมดทุกครั้ง" นี่จะช่วยแก้ปัญหาในเรื่องการทำให้หน้าจอเลอะเทอะโดยยูสเซอร์ไปด้วย กล่าวคือ ในงานที่มีการอินพุตข้อมูลที่เป็น Text นั้น ถ้าเราใช้คำสั่ง READ ตามธรรมดาก็จะเปิดโอกาสให้ยูสเซอร์จิ้มอะไร ๆ เข้ามาได้หมด สิ่งที่อินพุตเข้ามาอาจล้ำออกไปนอกช่องที่จะให้กรอกข้อมูล หน้าจอภาพจึงเลอะเทอะได้ เราจึงควรกำหนดให้มีการเคลียร์หน้าจอและเริ่มต้นกันใหม่ทุกครั้งที่อัพเดตเสร็จ

แนวทางในการออกแบบโปรแกรม

ก่อนอื่นเราคงต้องทำความเข้าใจกับเรื่องของการกำหนดเวลาบนพื้นผิวของโลกเราเสียก่อน ถ้าเราเอาแผนที่โลกมาดูก็จะพบว่าส่วนใหญ่เขาจะเอาเมืองกรีนิช (Greenich) ของประเทศอังกฤษเป็นศูนย์กลาง ส่วนของพื้นผิวโลกที่อยู่ตรงข้ามกับเมืองกรีนิชนั้นจะเป็นมหาสมุทรแปซิฟิกพอดี แผนที่โลกของเราจึงเหมือนกับการเอามีดกรีดลงไปบนมหาสมุทรแปซิฟิก โดยกรีดจากขั้วโลกเหนือลงมาสู่ขั้วโลกใต้ กรีดแล้วก็แบะมันออกเหมือนกับผิวผลส้ม แล้วก็ต้องตีแผ่ส่วนบนกับส่วนล่างให้ยืดออกจนเป็นแผ่นกระดาษ 4 เหลี่ยมผืนผ้า แนวที่เราเอามีดกรีดลงไปนั้นเขาเรียกว่า "เส้นแบ่งวัน" ถ้าเราจะลองสมมติว่าเส้นแบ่งวันนี้เป็นเหมือนเส้นผมของคนเราที่เอามาต่อ ๆ กันและมองไว้จากขั้วโลกเหนือไปจนจรดขั้วโลกใต้ก็อาจเข้าใจได้ดียิ่งขึ้น การกรีดลูกโลกและแบะออกเป็นแผ่นนี้ก็จะทำ

ให้เส้นผมดังกล่าวถูกผ่ากลางโดยตลอด แล้วส่วนหนึ่งก็จะติดไปอยู่ที่ขอบแผนที่โลกด้านซ้ายมือ และอีกส่วนหนึ่งก็จะติดไปอยู่ที่ขอบด้านขวา เจ้าเส้นผมดังกล่าวนี้จะเป็นวันเดียวกันแต่คนละเวลา คือด้านซ้ายมือจะเป็นเวลา 0.00 น. ส่วนด้านขวาจะเป็น 24.00 น. เมื่อพระอาทิตย์ไปส่องอยู่ตรงหัวของเมืองกรีนิชพอดี หรือคือเมืองกรีนิชเป็นเวลา 12.00 น. เจ้าเส้นผมด้านซ้ายมือจะเป็น 0.00 น. และด้านขวาจะเป็น 24.00 น. จากจุด 3 จุดดังกล่าวนี้เราสามารถจะคำนวณได้ว่าเมืองอื่น ๆ จะเป็นเวลาเท่าใด เมืองใดอยู่ห่างไปทางขวามือของกรีนิชจะมีเวลาสูงกว่าเมืองกรีนิช เมืองใดอยู่ไปทางซ้ายมือของกรีนิชก็จะมีเวลาต่ำกว่าเมืองกรีนิช จำนวนชั่วโมง นาที และวินาทีที่ต่างกันนี้คำนวณได้จากระยะทางตามแนวราบ วิธีการที่แยบยลกว่านั้นก็คือ ก่อนที่เราจะกรีดและแบะพื้นผิวลูกโลกออกมานั้น เขาให้ทำการลากเส้นจากขั้วโลกเหนือไปสู่ขั้วโลกใต้เอาไว้ เส้นเหล่านี้เขาให้มันห่างกัน 1 ส่วน 24 ของเส้นรอบวงที่เส้นศูนย์สูตรพอดี เส้นเหล่านี้เราเรียกว่า "เส้นแวง" มันจะทำมุมกันที่แกนของลูกโลกเป็นมุม 15 องศาพอดี ที่เป็นเช่นนี้ก็เพราะ 1 รอบโลกจะมี 360 องศา เมื่อเราแบ่งเป็น 1 ส่วน 24 ก็จะได้ส่วนละ 15 องศา เจ้าเส้นแวงเหล่านี้เมื่อเราแบะและตีแผ่พื้นผิวโลกออกมาจนเป็นรูปสี่เหลี่ยมผืนผ้าก็จะกลายเป็นเส้นที่วิ่งจากขอบบนลงมาสู่ขอบล่างเหมือนกันหมด แต่จะวางขนานกันและห่างกัน 1 ส่วน 24 ของเส้นศูนย์สูตรพอดี เจ้าเส้นแวงที่วิ่งผ่าเมืองกรีนิชนั้นเขาเรียกกันว่าเส้นที่ 0 องศา เส้นที่อยู่ถัดไปทางซ้ายก็จะเรียกว่าเส้นที่ 15, 30, 45,... องศาเช่นกัน และเพื่อให้รู้กันว่าเรากำลังจะพูดถึงเส้นไหนเขาก็บอกว่าให้เติมคำว่า "East of Greenich" และ "West of Greenich" เข้าไปด้วย สำหรับเมืองไทยของเรานั้นจะอยู่ระหว่างเส้นแวงที่ "90 East" กับ "105 East" หรือคือมีเวลาที่เดินเร็วกว่ากรีนิชประมาณ 6 ถึง 7 ชั่วโมง

ด้วยการมองดูลูกโลก ณ จังหวะที่พระอาทิตย์อยู่ตรงหัวของคนเมืองกรีนิช แล้วก็จดบันทึกเวลาของแต่ละเมืองเอาไว้เป็นตารางอยู่ในจาน

ชื่อเมืองและประเทศ STRING [17]	Local Flag	Greenich Mean Time HHMM	Daylight Saving Data			
			Flag X	From DDMM	To DDMM	shift ±MMM

Column	Type	Context
1-18	STRING[17]	ชื่อเมืองและประเทศ
19	BYTE	flag เพื่อบอกว่าเป็น local city หรือ City ธรรมดา หรือ Empty
20-24	STING[4]	เวลามาตรฐาน (เมื่อกรีนิช = เที่ยงวัน)
25	BYTE	flag บอกมีการใช้ Daylight Saving Time หรือไม่
26-30	STRING[4]	วันเริ่มใช้ Daylight Saving Time
31-35	STRING[4]	วันสิ้นสุด Daylight Saving Time
36-40	STRING[4]	จำนวนนาทีที่เพิ่มหรือลดในช่วง Daylight Saving Time

รูปที่ 2 แสดงรูปร่างของข้อมูลในแต่ละเรคอร์ด

แม่เหล็ก เราก็สามารถจะบอกได้ว่าเมืองไหนจะมีเวลาแตกต่างจากเมืองอื่นเป็นจำนวนเท่าใด นี่เป็นตัวเลขบอกเวลาในทางทฤษฎี คือถ้าห่างกัน 15 องศาก็จะต่างกัน 1 ชั่วโมงพอดี แต่ในทางปฏิบัตินั้นเราจะพบว่าเราทำเช่นนั้นไม่ได้ อย่างเช่น คนอยู่อำเภอพระโขนงจะเห็นพระอาทิตย์ช้ากว่าคนอยู่อำเภอมีนบุรีอยู่ 1-2 นาที ถ้าเราเอาตัวเลขทางทฤษฎีมาใช้ผมกับท่านผู้อ่านก็จะตั้งนาฬิกาไม่ตรงกัน คนในประเทศที่ไม่ใหญ่อย่างเราจึงต้องเลือกเอาค่าใดค่าหนึ่งมาใช้ เช่นเมืองไทยเราเลือกที่จะบอกว่าเราอยู่ห่างจากกรีนิชมา 7 ชั่วโมงพอดี ประเทศอื่น ๆ ก็ต้องเลือกเหมือนกัน พวกเขาจะเลือกให้ลงตัวเป็นชั่วโมงหรือครึ่งของชั่วโมง ยกตัวอย่างเช่น เมืองบอมเบย์ของประเทศอินเดียเขาเลือกว่าอยู่ห่างกรีนิชมาทางตะวันออก 5 ชั่วโมง 30 นาที นี่ผมหมายรวมถึงเมืองอีกจำนวนหนึ่งที่อยู่ในบริเวณใกล้เคียงกันด้วย สำหรับประเทศสหรัฐอเมริกานั้นเขามีพื้นที่กว้างใหญ่ไพศาล มันครอบคลุมห่างกันถึง 4 ชั่วโมง เขาจึงแบ่งประเทศออกเป็น 4 โซน แต่ละโซนใช้เวลาที่แตกต่างกันอยู่ 1 ชั่วโมงถัด ๆ กันไป ในสมัยที่ผมยังเป็นเด็กนักเรียนที่ได้ไปเรียนเมืองอเมริกาเรายังเข้าใจสับสนอยู่บ่อย ๆ เช่น เพื่อนที่อยู่ที่แอลเอเขาโทร.มาบอกว่าวันทอาทิตย์หน้าจะโทรมาใหม่ตอน 6 โมงเย็น แล้วเราก็มานั่งรอที่เครื่องโทรศัพท์ตอน 6 โมงเย็น รอเท่าไหร่ ๆ ก็ไม่โทร.มาสักที ผมก็เลยออกไปกินข้าว แต่พอกลับมา 3 ทุ่มเขาจึงโทร.มาถึง ที่เป็นเช่นนี้ก็เพราะผมอยู่เมืองแอตแลนต้าซึ่งจะเห็นพระอาทิตย์เร็วกว่าเมืองลอสแองเจลีสอยู่ประมาณ 3 ชั่วโมง

มัวไปฟุ้งถึงเรื่องการกำหนดเวลาของประเทศต่าง ๆ อยู่นาน โดยสรุปก็คือ **ในแต่ละเรคอร์ดเราจะเก็บตัวเลขบอกเวลา ณ จังหวะที่พระอาทิตย์ตรงหัวของชาวเมืองกรีนิชเอาไว้** เช่นประเทศไทยเราทั้งประเทศจะเป็นตัวเลข = 12:00+7:00 = 19:00 น. เมืองบอมเบย์ก็จะเป็นตัวเลข 12:00+5:30 = 17:30 น. เมืองนิวยอร์คก็จะเป็น 12:00 - 5:00 = 7:00 น.,... ฯลฯ เมื่อเข้าใจตรงนี้แล้วก็จะสบายบรื๋อ เจ้าเรคอร์ดของผมจึงมีหน้าตาดังแสดงในรูปที่ 2

เห็นหน้าตาของเรคอร์ดแล้วท่านผู้อ่านอาจสงสัยว่าเจ้าข้อมูลเกี่ยวกับ Daylight Saving Time นั้นคืออะไร เรื่องนี้บอกสั้น ๆ ว่าประเทศที่อยู่ห่างเส้นศูนย์สูตรนั้นความยาวของกลางวันจะสั้นลงกลางคืนจะยาวขึ้น นี่เป็นเพราะแกนของโลกเราทำมุมไม่ตั้งฉากกับแนวโคจรรอบดวงอาทิตย์ บางประเทศจึงประกาศเปลี่ยนแปลงเวลามาตรฐานไปจำนวนหนึ่ง โดยมากจะเป็น 1 ชั่วโมง เมื่อออกจากฤดูดังกล่าวแล้วก็จะกลับมาใช้เวลามาตรฐานเดิม เรื่องนี้ผมใส่เตรียมเอาไว้สำหรับงานในอนาคต ตอนนี้จะยังไม่ใช้งาน

จากรูปร่างของเรคอร์ดเราก็มาวางแผนในการโปรเซสข้อมูล วิธีการอันหนึ่งที่ผมอยากจะแนะนำให้ใช้ก็คือ เราจะอ่านข้อมูลจากไฟล์ขึ้นมาเก็บไว้ในเมมโมรีให้หมด แล้วใช้ข้อมูลในเมมโมรีในการทำงานตามปกติ เมื่อไหร่ที่มีการปรับปรุงแก้ไขข้อมูล เราก็จะทำกับข้อมูลในเมมโมรีเสียก่อน เสร็จแล้วจึงจะก๊อบปี้ลงไปใส่ในไฟล์ เราจะก๊อบปี้ลงไปทั้ง 200 เรคอร์ดเลยทีเดียว ไฟล์นี้เราจะเปิดใช้แต่ในตอนที่โหลดข้อมูลเข้าเมมโมรีและในตอนก๊อบปี้ลงเท่านั้น ทำแต่ละครั้งเสร็จแล้วให้ปิดไฟล์ได้เลย

ด้วยวิธีนี้จะช่วยให้การอัพเดตทำได้เร็ว และจะมีโอกาสสูญเสียข้อมูลจากการเกิดไฟฟ้าดับได้น้อยมาก ทั้งนี้เพราะไฟล์จะถูกเปิดใช้ในช่วงเวลาสั้น ๆ (less than 1 second each) เท่านั้น โอกาสที่ไฟฟ้าจะมาดับเอาตอนช่วงเวลาดังกล่าวมีน้อยมาก เทคนิคนี้จะดีกว่าการเปิดไฟล์เอาไว้ตลอดเวลา ซึ่งเป็นเทคนิคที่พวกเราถูกสอนให้ใช้กันโดยทั่วไป

ข้อดีอีกประการหนึ่งของการก๊อบปี้ข้อมูลทั้งไฟล์มาไว้ในเมมโมรีก็คือ การทำงานของโปรแกรมจะเร็วขึ้นมาก อย่างเช่น จากตารางข้อมูลที่เก็บอยู่ในเมมโมรี ซึ่งในโปรแกรมของผมให้เรียกว่า CITYTAB (คำนี้ย่อมาจากคำว่า CITY-RECORD-TABLE) เราสามารถตั้งตัวแปรขึ้นมา 5 ตัว คือ LOCALPOSITION, FOREIGNPOSITION, CURSORPOSITION, DISPLAYFROM และ MAXPOSITION เจ้าตัวแรกนั้นชี้ไปยัง CITYTAB ตัวที่เก็บโลคัลซิตี้ ตัวที่ 2 ชี้ไปยัง CITYTAB ตัวที่เก็บฟอรีนซิตี้ที่กำลังโชว์เวลาอยู่บนจอภาพ ตัวที่ 3 บอกตำแหน่งที่อยู่ของเคอร์เซอร์ ตัวที่ 4 บอกตำแหน่งเริ่มต้นของชื่อเมืองที่อยู่บนจอภาพ (ตัวที่ชี้ไปยังเมืองสุดท้ายบนจอภาพนั้นไม่จำเป็นต้องมี) ส่วนตัวสุดท้ายจะชี้ไปยังเรคอร์ดสุดท้ายที่มีข้อมูลอยู่จริง ด้วยการมองให้เห็นว่าเรากำลังกระทำกับ CITYTAB อันเดียวอยู่ตลอดเวลา เราจะไม่ต้องกังวลกับเรื่องจานแม่เหล็กเลย เช่น การอัพเดตแบบเพิ่มชื่อเมือง (=ADD) เราสามารถจะโยกย้ายหรือเลื่อนข้อมูลใน CITYTAB ไปมาได้อย่างรวดเร็ว เมื่อนำข้อมูลใหม่แทรกลงได้ถูกที่ (เรียงตามตัวอักษร) ได้แล้ว ก็ไปเรียกโพรซีเดอร์ UPDATECITYFILE มาทำงาน โพรซีเดอร์อันนี้จะเปิดไฟล์ ก๊อบปี้จาก CITYTAB ลงไปแล้วก็เปิดไฟล์ไปเลย ทั้งหมดนี้จะทำงานได้เสร็จในเวลาไม่ถึงเสี้ยวของวินาที ในการสั่งลบข้อมูล (DELETE) ก็เหมือนกัน เราทำกับ CITYTAB เสียก่อน แล้วจึงเรียกใช้ UPDATECITYFILE

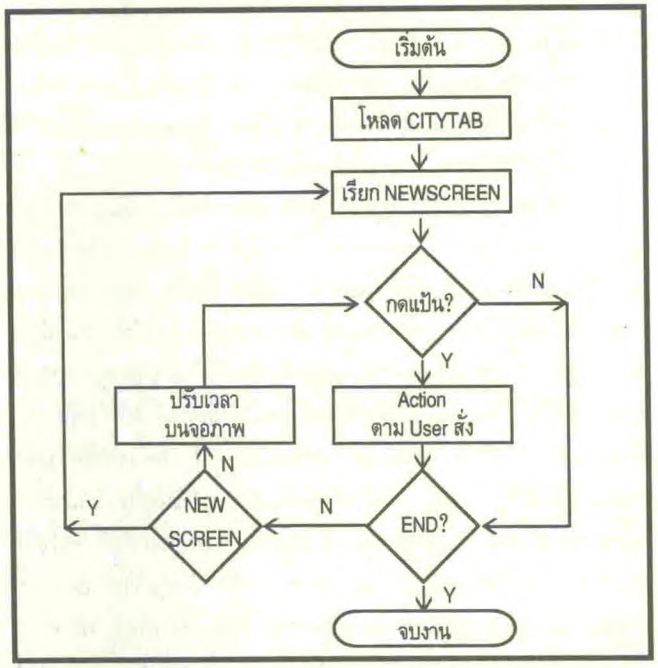
โลจิกของโปรแกรมนี้จะเป็นดังแสดงในรูปที่ 3 ส่วนรูปที่ 4 นั้นแสดงไดอะแกรมของ CITYTAB

รายละเอียดบางอย่างในโปรแกรม

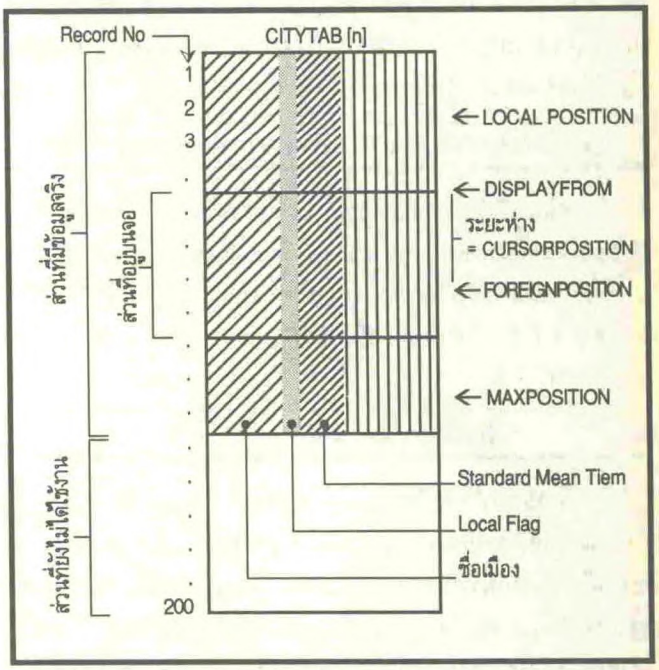
จริง ๆ แล้วโปรแกรมนี้จะประกอบไปด้วยโพรซีเดอร์หรือรูทีนที่สำคัญ ๆ 18 รูทีน ซึ่งผมจะขออธิบายไปทีละตัวดังนี้

GETDATE นี่เป็นรูทีนที่เราจะติดต่อกับดอสเพื่อขอวันเดือนปี วิธีการขอเทอร์โบปาสคาลเวอร์ชัน 3.0 คือ เราจะต้องกำหนดเรคอร์ดขึ้นมาอันหนึ่ง ผมเรียกมันว่า REG1 เรคอร์ดนี้ประกอบไปด้วยตัวแปรชื่อ AL, AH, BL, BH, CX, DL, DH ตามลำดับ เจ้า AH คือรหัสการอินเตอร์รัพต์ดอส รหัสการของอินเตอร์รัพต์เพื่อขอวันเดือนปี คือ $2A (เจ้าเครื่องหมาย $ นี้หมายถึงว่าสิ่งที่ตามมาจะเป็นค่าแบบ Hexa Decimal) เราต้องเอาค่านี้ใส่เข้าไปที่ AH เสียก่อน เสร็จแล้วก็เรียกหาดอสโดยคำสั่ง MSDOS(REG1); เมื่อดอสได้รับแล้วก็จะเอาข้อมูลวันเดือนปีมาใส่ในตัวแปรอื่น ๆ ให้เรา เช่น เอาค่ารหัสของวัน (0= อาทิตย์, 1 = จันทร์, 2= อังคาร, ...) มาใส่ให้ที่ AL เอาค่าของปีมาใส่ให้ที่ CX เอารหัสของเดือน (01 = มกราคม, 02= กุมภาพันธ์,...) มาใส่ให้ที่ DH และเอารหัสวันที่ (01, 02,...31) มาใส่ให้ที่ DL เมื่อผ่านการติดต่อกับดอสแล้วเราก็ย้ายค่าเหล่านี้มาใส่ DAY, YEAR, MONTH และ DATE ของเราได้ หลังจากนั้นเราก็เอาค่าเหล่านี้ไปใช้คำนวณและแสดงผลบนจอภาพต่อไป

GETTIME นี่เป็นรูทีนที่คล้ายกับอันข้างบน เราต้องใส่รหัสอินเตอร์รัพต์ $2C ถ้าไปที่ AH เมื่อติดต่อกับดอสโดยคำสั่ง MSDOS



รูปที่ 3 โครงสร้างของโปรแกรม



รูปที่ 4 โครงสร้างของข้อมูลในเมมโมรี

(REG2); แล้วก็จะได้ตัวเลขบอกชั่วโมงอยู่ใน CH ตัวเลขบอกนาทีอยู่ใน CL ตัวเลขบอกวินาทีอยู่ใน DH และตัวเลขเศษของวินาทีอยู่ใน DL เรื่องของการติดต่อกับดอสนี้เขามีรายละเอียดอยู่ในคู่มือของ DOS เขาเรียกกันว่า Dos Function Call ท่านผู้อ่านควรไปหามาอ่านดู การเล่นกับ DOS Function นี้ก็เป็นเรื่องน่าสนุกอย่างหนึ่ง คู่มือที่เป็นภาษาไทยและเข้าใจได้ไม่ยากคือเล่มที่ชื่อว่า "คู่มือ MS-DOS และ PC-DOS" ของอาจารย์วศิน เพิ่มทรัพย์ เรื่องนี้ผมจะหาโอกาสทดลองและนำมาเผยแพร่ในวันหลัง

DRAWLOCALTIME นี้เป็นรูทีนที่จะเอาข้อมูลที่ได้จากดอสมาแสดงบนหน้าจอภาพ ผู้อ่านควรสังเกตว่า DAY, YEAR, MONTH และ DATE ที่เราได้มานั้นจะเป็นของโลคัลซิตี้ ทั้งนี้เพราะเป็นเวลาจากภายในเครื่อง คนที่ใช้เครื่องอยู่นั้นย่อมจะเซตค่าวันเดือนปีตามเมืองที่เขาอาศัยอยู่ ในกรณีที่ท่านย้ายไปอยู่เมืองอื่นหรือประเทศอื่น ท่านจะต้องเซตเวลาเครื่องเสียก่อนที่จะวิ่งไปโปรแกรมนี้ และสิ่งแรกที่คุณต้องทำเมื่อวิ่งโปรแกรมนี้ก็คือใช้คำสั่ง File Maintanance และเลือกฟังก์ชัน Set Local City ให้ตรงกับเมืองใหม่ที่ท่านไปอยู่ ในกรณีที่เมืองที่ท่านย้ายไปอยู่นั้นไม่มีอยู่ในไฟล์ ท่านก็จะต้องทำฟังก์ชัน Add เสียก่อน ขอให้สังเกตว่ารูทีนนี้เป็นรูทีนที่ทำงานแบบตรงไปตรงมา คือได้ชื่อเมืองก็เอาไป WRITE ลงที่บรรทัดที่ 5 ได้รหัส DAY มาก็ convert ไปเป็น SUNDAY, MONDAY,... โดยรูทีน CONVERTDAY (LOCALDAY, LOCALDAYNAME); แล้วก็เอาไป WRITE ลงบรรทัดที่ 7 ...ฯลฯ ที่สำคัญมีอยู่อย่างเดียวคือ มีการคำนวณค่า LOCALTIME และ LOCALMEANTIME เอาไว้ หน่วยจะเป็น "นาที" ค่านี้จะต้องนำไปใช้ในการคำนวณวันเดือนปีและเวลาของฟอรีนซิตี้

DRAWFOREIGNTIME นี่เป็นรูทีนที่ต้องคิดมากหน่อย มันจะเอาชื่อเมืองไปแสดงบนจอภาพก่อน เสร็จแล้วไปดึงเอา MEANTIME จาก CITYTAB [FOREIGNPOSITION] ออกมา มันจะคำนวณเป็น "นาที" เสร็จแล้วมาเข้าสมการว่า

```
FOREIGNTIME-LOCALTIME = FOREIGNMEANTIME-LOCALMEANTIME
```

สมการนี้เป็นจริงตลอดเวลา ทั้งนี้เพราะไม่ว่าโลกจะหมุนไปอยู่ในตำแหน่งใด ผลต่างของเวลาของสองเมืองนี้จะเท่าเดิม ผลต่างทางด้านขวามือของสมการนั้นได้มาจากข้อมูลตารางเวลามาตรฐานที่กล่าวมาในตอนต้น ค่านี้เป็นค่าคงที่ ผมให้เรียกว่า TIMEDIFF จากสมการนี้เราก็จะได้ผลว่า

```
FOREIGNTIME = TIMEDIFF + LOCALTIME
```

สิ่งที่ควรสังเกตในเรื่องนี้ก็คือ ค่าของ FOREIGNTIME ที่คำนวณมาได้นี้อาจมีค่ามากกว่า 60x24 = 1,440 นาที หรือเป็นลบไปเลยก็เป็นได้ ในกรณีที่ FOREIGNTIME มีค่าอยู่ระหว่าง 0 กับ 1,440 จะแสดงว่าฟอรีนซิตี้ยังอยู่ในวันเดียวและปีเดียวกันกับโลคัลซิตี้ แต่ถ้า FOREIGNTIME มีค่ามากกว่าหรือเท่ากับ 1,440 นาที มันจะบอกให้เราทราบว่าฟอรีนซิตี้นั้นอยู่ในวันเดือนปีที่ล้ำหน้าไป 1 วัน ในกรณีนี้เราจะต้องคำนวณวันเดือนปีกันใหม่ มันจะแตกเป็นกรณีย่อยได้ 3 กรณี คือ ถ้าโลคัลซิตี้อยู่ ณ วันที่ 31 ธันวาคม เราจะต้องขึ้นปีใหม่ให้แก่ฟอรีนซิตี้ กรณีที่ 2 คือโลคัลซิตี้อยู่ตรงกับปลายเดือนของมกราคมหรือกุมภาพันธ์... พฤศจิกายน เราก็จะต้องขึ้นเดือนใหม่ให้กับฟอรีนซิตี้ ส่วนกรณีสุดท้ายคือโลคัลซิตี้อยู่ยังไม่ถึงวันสิ้นเดือน กรณีนี้เราแค่เพิ่มวันที่ไปอีก 1 วันเท่านั้น โปรแกรมส่วนนี้เลยต้องมี IF Statement อยู่เป็นตับไปเลย หลังจากปรับวันเดือนปีไปแล้วเราก็ต้องเอา 1,440 มาลบออกจาก FOREIGNTIME ด้วย นั่นจึงจะเป็นตัวเลขบอกเวลาจริง สำหรับกรณีที่ FOREIGNTIME มีค่าเป็นลบนั้นมันจะแสดงว่าฟอรีนซิตี้นั้นจะอยู่ในวันที่ย้อนไป 1 วัน นี่ก็เหมือนกัน คือต้องเช็คว่าโลคัลซิตี้อยู่ ณ วันที่ 1 ของเดือนมกราคมหรือไม่ ถ้าเข้ากรณีนี้เราต้องปรับทั้งวัน เดือน และปี ถ้าไม่ใช่เราก็ต้องเช็คว่าโลคัลซิตี้นั้นอยู่ ณ วันที่ 1 ของเดือนกุมภาพันธ์หรือมีนาคมหรือ...ธันวาคมหรือไม่ ถ้าใช่ก็ต้องปรับเดือนลงมา 1 เดือน และปรับวันที่เป็นวันสิ้นเดือนของเดือนที่ปรับแล้ว ส่วนในกรณีที่โลคัลซิตี้ไม่ได้อยู่ ณ วันที่ 1 เราก็แค่ปรับลดวันที่ลงมา 1 วัน หลังจากปรับแก้วันเดือนและปีเสร็จแล้วเราก็ยังจะต้องเอาเลข 1,440 มาบวกเข้าที่ FOREIGNTIME นี่จึงจะเป็นเวลาที่จะใช้แสดงบนจอภาพ

รูทีนเพื่อการแสดงวันเดือนปีและเวลาของฟอรีนซิตี้นี้ยุ่งยากพอสมควร เขียนเสร็จแล้วก็ต้องทดสอบให้ดี ๆ อนึ่งท่านจะต้องระวังเรื่องเดือนกุมภาพันธ์ ซึ่งอาจมี 28 หรือ 29 วันก็เป็นได้ การปรับวันเดือนปีขึ้นหรือลงจะต้องเช็คเรื่องนี้เอาไว้ด้วย

DISPLAYCITYNAMES นี่เป็นรูทีนสำหรับเอาชื่อเมืองจำนวน 11 ชื่อขึ้นไปแสดงบนจอภาพ เรื่องนี้เราทำได้โดยไม่ต้องแคร์ว่าจะเป็นเรคอร์ดที่มีข้อมูลหรือไม่มีข้อมูล ในกรณีที่ไม่มีข้อมูลมันก็จะเป็น blank อยู่แล้ว ทั้งนี้เพราะเวลาเราสร้างไฟล์และอัพเดตไฟล์นั้นเราจะต้องเคลียร์บริเวณที่เป็นชื่อเมืองทิ้งไปอยู่แล้ว จุดที่เราต้องระมัดระวังก็คือก่อนที่จะสั่ง WRITE บนหน้าจอเราจะต้องเช็คเสียก่อนว่ามันตรงกับตำแหน่งที่เป็นฟอรีนซิตี้หรือไม่ ถ้าใช่ก็ให้ใช้สีแดง (โดยใช้คำสั่ง TEXTCOLOR(2);) ส่วนเมืองอื่นให้ใช้สีขาว (โดยคำสั่ง TEXTCOLOR(3);)

DISPLAYOPTIONS นี่เป็นรูทีนเพื่อแสดงข้อความบนจอภาพเพื่อให้ยูสเซอร์รู้ว่าเขาเลือกกดแป้นอะไรได้บ้าง ในโปรแกรมนี้ผมบังคับว่าจะต้องอินพุตเป็น Capital Letter (ตัวใหญ่) ทั้งนั้น และแป้นต่าง ๆ ที่กดได้จะมีดังนี้คือ E= จบโปรแกรม U= เลื่อนเคอร์เซอร์ขึ้น D= เลื่อนเคอร์เซอร์ลง F= ขอปรับแก้ข้อมูลในแฟ้มข้อมูล ในกรณีที่ยูสเซอร์กดแป้น F เราก็จะขึ้น option หรือฟังก์ชันย่อยให้เลือกได้ 3 อย่างคือ A= เพิ่มเมือง D= ลบเรคอร์ดที่เป็นฟอรีนซิตี้ในขณะนั้น S= เซตฟอรีนซิตี้ในขณะนั้นให้เป็นโลคัลซิตี้ (โลคัลซิตี้เก่าก็จะกลายเป็นเมืองธรรมดาเมืองหนึ่ง) ท่านผู้อ่านควรสังเกตว่าผมพยายามสร้างรูทีนย่อย ๆ ที่เข้าใจได้ง่าย ๆ และจะให้ผลที่แน่นอนเอาไว้แยะ ๆ เสร็จแล้วค่อยไปเรียกใช้ในเมนโปรแกรมหรือโพรซีเดอร์หลักอื่น ๆ) วิธีนี้จะทำให้ดีบักได้ง่าย

NEWSCREEN เป็นรูทีนเพื่อเขียนหน้าจอที่เป็นพื้นฐานและการเซตค่าต่าง ๆ เช่น การตีกรอบ การโหลดข้อมูล จาก CITYFILE ลง

CITYTAB การค้นหา LOCALPOSITION การเซตค่า FOREIGNPOSITION การเซตค่า CURSORPOSITION ... และในขั้นสุดท้ายก็ไปเรียกใช้รูทีน DRAWLOCALTIME, DRAWFOREIGNTIME, DISPLAYCITYNAMES และ DISPLAYOPTIONS ออกมาใช้งาน ภาพบนจอภาพก็จะครบถ้วน รูทีน NEWSCREEN นี้จะเคลียร์ภาพเก่าทิ้งหมดด้วย ดังนั้นมันจึงมีประโยชน์ในกรณีที่เราสงสัยว่าหน้าจอของเราจะถูกยูสเซอร์ทำเลอะเทอะด้วย

UPDATETIMECOUNT นี่เป็นรูทีนที่จะใช้เปลี่ยนแปลงเวลาบนจอภาพ ในรูทีนนี้เราจะเรียกใช้ GETTIME และ GETDATE เสียก่อน เมื่อได้วันเวลาแล้วเราลองเช็คดูว่ามีการเปลี่ยนแปลงไปจากที่แสดงบนจอภาพหรือไม่ ค่าเก่าจะเก็บไว้ที่ OLDHOUR, OLDMIN, OLDSEC ถ้ายังเป็นเวลาเท่าเดิมเราก็ไม่ต้องทำอะไร แต่ถ้าเวลาใหม่ไม่ตรงกับเวลาเก่าเราก็ต้องเรียกใช้ DRAWLOCALTIME และ DRAWFOREIGNTIME เสร็จแล้วก็ต้องเอา HOUR, MIN และ SEC อันใหม่ใส่ไปไว้ใน OLDHOUR, OLDMIN และ OLDSEC ไว้ด้วย รูทีน UPDATETIMECOUNT นี้จะถูกเรียกใช้ในลูปของ main program

CREATECITYFILE, LOADCITYTABLE, UPDATECITYFILE รูทีนเหล่านี้ใช้เพื่อการสร้างไฟล์ เพื่อการโหลดข้อมูลจากไฟล์ลงไป CITYTAB และเพื่อการก๊อบปี้ข้อมูลจาก CITYTAB ลงในไฟล์ รูทีนเหล่านี้สั้น ง่าย และเชื่อได้ว่าจะทำงานได้ตามที่เราตั้งใจไว้ ผมคิดว่าท่านผู้อ่านคงจะอ่านจากลิสติงของโปรแกรมและทำความเข้าใจได้

MOVECURSORUP, MOVECURSORDOWN เป็นรูทีนสำหรับการเลื่อนแถบสี (=Cursor) ขึ้นและลงตามลำดับ รูทีนทั้งสองดูง่าย ๆ แต่มีหลักการที่สำคัญอยู่ภายใน กล่าวคือเราจะเช็คดูเสียก่อนว่าเราจะเลื่อนรายชื่อขึ้นหรือลงได้อีกหรือไม่ ถ้าทำได้ก็ให้ปรับค่า FOREIGNPOSITION เอาไว้ก่อน เสร็จแล้วค่อยมาเช็คว่าต้องปรับ CURSORPOSITION ตามไปด้วยหรือไม่ ถ้าปรับได้ก็ปรับ แต่ถ้าปรับไม่ได้มันก็แสดงว่าเคอร์เซอร์ขึ้นไปอยู่ติดขอบบนหรือขอบล่างแล้ว กรณีนี้ก็ไม่ต้องปรับ แต่อย่างไรก็ตาม ทั้งสองกรณีเราจะต้องเรียกรูทีน DISPLAYCITYNAMES มาใช้ทุกครั้ง วิธีการนี้อาจทำให้เครื่องต้องทำงานมากหน่อย แต่มันก็ใช้เวลาเพียงเสี้ยวของวินาทีมันจึงไม่มีผลในแง่ของการใช้งาน

ADDTOTABLE, DELETEFROMTABLE, SETLOCALCITY 3 รูทีนนี้ใช้เพื่อการอัพเดตไฟล์ จุดที่ต้องระมัดระวังในรูทีนทั้ง 3 ก็คือ เราจะต้องรักษาการจัดเรียงตามตัวอักษรเอาไว้อยู่เสมอ นอกจากนี้แล้วเรคอร์ดที่ถูกเลิกใช้จะต้องเคลียร์ทิ้งและให้พร้อมที่จะถูกเรียกมาใช้เก็บข้อมูลได้อีก ข้อควรสังเกตอีกประการหนึ่งก็คือ การอัพเดตเหล่านี้จะมีผลกระทบต่อไฟล์และสถานะของเรคอร์ดต่าง ๆ ดังนั้นหลังจากที่เราก๊อบปี้ CITYTAB ลงไปไว้ในไฟล์แล้วก็จะต้องเรียกใช้ NEWSCREEN ทันที

FILEMAINTENANCE, PROCESSKEY 2 รูทีนนี้ใช้เพื่อจัดการกับคำสั่งของยูสเซอร์ โปรแกรมของเราจะคอยเช็คอยู่เสมอ ๆ ว่ายูสเซอร์ได้กดแป้นคีย์บอร์ดหรือไม่ คำสั่งนี้คือ IF KEYPRESSED THEN...ซึ่งอยู่ในส่วนที่เป็น Main Part ถ้ายูสเซอร์ไม่ได้กดแป้นคีย์บอร์ดโปรแกรมก็จะวนอยู่ในลูปต่อไป

ผมคิดว่าผมได้อธิบายรายละเอียดมาพอสมควรแล้ว ถึงตอนนี้ท่านก็ควรจะลองข้ามไปอ่านโปรแกรมดู ถ้ายังไม่ค่อยจะเข้าใจก็ลองติดต่อสำนักพิมพ์เพื่อขอก๊อบปี้โปรแกรมไปเล่น ในดิสเกตต์จะประกอบไปด้วย source program ชื่อว่า ATMSOI.PAS และ ATMSOI.BAK โปรแกรมนี้ใช้กับเทอร์โบปาสคาลเวอร์ชัน 3.0 นอกจาก sourcces program แล้วก็จะมี object program ที่ชื่อว่า ATMSOI.COM ท่านสามารถเรียกไปทำงานได้เลย สำหรับไฟล์ที่เก็บข้อมูลของเมืองต่าง ๆ นั้นใช้ชื่อว่า ATMDO2.DAT ท่านควรจะก๊อบปี้เก็บเอาไว้เสียก่อน ทั้งนี้เพราะถ้าท่านเล่นผิด ๆ ถูก ๆ มันอาจเสียหายได้ ยกตัวอย่างเช่น ใช้คำสั่ง DELETE ซ้ำหลาย ๆ หนก็อาจทำให้ข้อมูลลดน้อยลง แล้วท่านก็ไม่สามารถสร้างใหม่ได้ ทั้งนี้เพราะท่านอาจไม่มีแผนที่โลกและตารางเวลามาตรฐานอยู่ในมือ เมื่อพูดถึงเรื่องตารางเวลามาตรฐานแล้วก็นึกขึ้นมาได้ คือผมควรจะขออนุญาตขอบคุณต่อผู้จัดพิมพ์หนังสือชื่อ "Thailand Airline Timetable" เอาไว้ในที่นี้ด้วย หนังสือนี้เข้าใจว่าจัดพิมพ์โดยบริษัทหรือสมาคมทางด้านการบินพาณิชย์ระหว่างประเทศ ในหนังสือเล่มนี้เขามีตารางเวลามาตรฐานของเมืองต่าง ๆ กว่า 150 เมือง ผมได้อาศัยหนังสือดังกล่าวมาใช้ในการสร้าง CITYFILE และเมื่อทดสอบแล้วก็คิดว่าถูกต้องเล่มที่ผมใช้อยู่นั้นเป็นของเดือน November 1996 ถ้าท่านผู้อ่านสามารถจัดหามาได้ (ราคาประมาณ 50 บาท) ก็สามารถจะเล่นอย่างไรก็ได้ ถ้าไฟล์มันพังไปก็สร้างขึ้นใหม่ได้

วิเคราะห์และวิจารณ์ผลงาน

โดยปกติแล้วเมื่อผมเขียนโปรแกรมเสร็จแล้วก็จะทดลองใช้เล่นไปสักระยะหนึ่ง ในช่วงเวลาดังกล่าวนี้ผมจะพยายามเก็บข้อดีและข้อเสียของโปรแกรมนั้นเอาไว้ มันเป็นเครื่องเตือนใจสำหรับงานในอนาคต โปรแกรมตัวอย่างที่ผมทำขึ้นมาในครั้งนี้ผมพบจุดดีและจุดเสียดังต่อไปนี้

1. โปรแกรมยังมีบักเหลืออยู่ 2-3 แห่ง แห่งแรกนั้นอยู่ในโพรซีเดอร์ DRAWFOREIGNTIME มันอยู่ใน statement ที่เขียนว่า IF FOREIGNTIME > 60*24 THEN... ที่เป็นการทดสอบว่าฟอรีนซิตี้มีเวลาเกิน 1 วันหรือไม่ เรื่องนี้จริง ๆ แล้วมันจะต้องเปลี่ยนเป็น IF FOREIGNTIME >= 60*24 THEN... ทั้งนี้เพราะในกรณีที่ FOREIGNTIME มีค่าเท่ากับ 60*24 นาทีนั้นก็เข้าข่ายที่จะต้องปรับวันเดือนและปีเช่นกัน การนับเวลาตามมาตรฐานจริง ๆ นั้นเขาเริ่มจาก 00:00:00 น. เมื่อครบ 1 วินาทีก็จะเปลี่ยนเป็น 00:00:01 น. มันจะนับไปเรื่อย ๆ จนถึง 23:59:59 และเมื่อครบ 1 วินาทีก็จะเปลี่ยนไปเป็น 00:00:00 น. ของวันใหม่ หรือคือ เวลา 24:00:00 น.นั้นไม่มีในระบบการนับเวลาที่แท้จริง ความละเอียดถี่ถ้วนพวกนี้เรามักจะหลงลืมอยู่เสมอ

2. มีเรื่องแปลกบางอย่างที่โจทย์ไม่ได้บอกไว้ คือประเทศนิวซีแลนด์กับประเทศออสเตรเลียนั้นเขาใกล้กัน และเขามีความสนิทชิดชอบกันอย่างแน่นแฟ้น เขาพยายามใช้อะไร ๆ ร่วมกันอยู่หลายอย่าง สิ่งหนึ่งก็คือ เขาพยายามจะใช้วันและเวลาที่ใกล้ชิดกัน แต่เมื่อเราดูแผนที่

โลกแล้วจะพบว่า เจ้าเส้นแบ่งวันมันไปผ่าระหว่าง 2 ประเทศนี้พอดี หรือคือ ประเทศนิวซีแลนด์นั้นจะไปอยู่ด้านซ้ายสุดของแผนที่ ส่วนประเทศออสเตรเลียนั้นไปอยู่ด้ายขวาสุดของแผนที่ ซึ่งถ้าดูกันตามแผนที่แล้วเวลามาตรฐานของเมือง Auckland ของประเทศนิวซีแลนด์ก็ควรจะเป็น -11 หรือ -12 ชั่วโมงนับจากเมืองกรีนิช และเวลาของเมือง Sydney ของประเทศออสเตรเลียก็ควรจะเป็น +11 หรือ +12 ชั่วโมงนับจากเมืองกรีนิช ปัญหาอันนี้ทำให้มีการแหกกฎไปเล็กน้อย คือเมือง Auckland นั้นเขาเลือกใช้เวลาเป็น +13 ชั่วโมง เมื่อเทียบกับกรีนิช หรือคือเวลามาตรฐานของเมือง Auckland จะเท่ากับ 12+13 = +25 ซึ่งมากกว่า 24 เรื่องนี้ทำให้เกิดปัญหา 2 ประการคือ เวลาที่เรารับข้อมูลจากยูสเซอร์เข้ามานั้นจะต้องยอมรับตัวเลขที่เกิน 24 ได้ด้วย ปัญหาประการที่ 2 คือ มันจะทำให้การคำนวณวันเดือนปีในรูทีน DRAWFOREIGNTIME ทำงานผิดพลาดได้

3. ในโปรแกรมตัวอย่างอันนี้ผมใช้โหมดของจอภาพเป็นแบบ CGA โหมดนี้จะแสดงตัวอักษรที่มีขนาดใหญ่ขึ้น ผมคิดว่ามันจะช่วยให้ดูเด่นชัดขึ้น แต่ทำไปแล้วจึงมาพบในภายหลังว่ามันไม่ช่วยอะไรเลย แถมยังทำให้ดูไม่ชัดเจน จำนวนตัวอักษรต่อ 1 บรรทัดมันลดลงเหลือแค่ 40 ตัว ชื่อเมืองกับประเทศจึงต้องตัดให้เหลือแค่ 17 ตัวอักษร ตัวชื่อประเทศจึงถูกตัดทิ้งไปบางส่วนเกือบทุกเมือง เรื่องนี้ช่วยสอนให้เรารู้ว่า "ตัวใหญ่" นั้นไม่ใช่สิ่งที่ดีเสมอไป ในเวอร์ชันที่จะใช้งานจริงคงต้องปรับไปใช้วินโดวส์และตัวอักษรที่มีทั้งตัวเล็กและตัวใหญ่

4. โปรแกรมตัวอย่างอันนี้นอกจากจะใช้ทำงานจริง ๆ ได้แล้ว ผมยังคิดว่าจะมีประโยชน์อีกหลายอย่าง เช่น ใช้สอนวิธีการติดต่อกับ DOS วิธีการจัดเรียงข้อมูลในเมมโมรี วิธีการเพิ่มความเร็วในการทำงานโดยใช้เมมโมรีแทนจานแม่เหล็ก วิธีการใช้ตัวแปร KEYPRESSED วิธีการลดความเสี่ยงของการสูญเสียข้อมูล

ผมเชื่อว่าโจทย์นี้และโปรแกรมตัวอย่างอันนี้คงจะเป็นประโยชน์กับท่านผู้อ่านพอสมควร ถ้าท่านอ่านแล้วยังไม่เข้าใจหรือสงสัยอะไรก็โทรไปคุยกันได้ที่ โทร. 311-4767 และบอกว่า "สมาชิกไมโครคอมพิวเตอร์ขอคุยกับอาจารย์ปัญญา..."

ลิสติงที่ 1 โปรแกรมตัวอย่าง

```
PROGRAM ATMS01;
{$I GRAPH.P }
LABEL NEWSCREENPOINT,MAINLOOP,EOJ;
TYPE  STR04=STRING[4];
      STR09=STRING[9];
      STR17=STRING[17];
      CITYRECTYPE=RECORD
        CITYNAME:STR17;
        LOCALFLAG:CHAR;
        MEANTIME:STR04;
        DAYLIGHTSAVINGFLAG:CHAR;
        DLSFROM,DLSTO:STR04;
        DLSSHIFT:STR04;
        END;
VAR CH:CHAR;
VAR CITYFILE:FILE OF CITYRECTYPE;
VAR CITYREC:CITYRECTYPE;
VAR CITYTAB:ARRAY[1..200] OF CITYRECTYPE;
VAR LOCALCITYNAME,FOREIGNCITYNAME:STR17;
VAR LOCALDAYNAME,FOREIGNDAYNAME:STR09;
VAR LOCALHOUR,LOCALMIN,FOREIGNHOUR,FOREIGNMIN:INTEGER;
VAR LOCALTIME,FOREIGNTIME:INTEGER;
VAR LOCALMEANTIME,FOREIGNMEANTIME:INTEGER;
VAR LOCALDAY,LOCALYEAR,LOCALMONTH,LOCALDATE:INTEGER;
VAR FOREIGNDAY,FOREIGNYEAR,FOREIGNMONTH,FOREIGNDATE:INTEGER;
VAR LOCALPOSITION,FOREIGNPOSITION:INTEGER;
VAR DISPLAYFROM,CURSORPOSITION,MAXPOSITION:INTEGER;
VAR OLDHOUR,OLDMIN,OLDSEC:INTEGER;
VAR ENDSWT,NEWSCREENSWT:INTEGER;
VAR HOUR,MIN,SEC,FRAG,SPF:INTEGER;
VAR DAY,YEAR,MONTH,DATE:INTEGER;

PROCEDURE GETDATE;
TYPE REGPACK=RECORD
          AL,AH,BL,BH:BYTE;
          CX:INTEGER;
          DL,DH:BYTE;
          END;
VAR REG1:REGPACK;
BEGIN
   WITH REG1 DO
     BEGIN
     AH:=$2A;
     MSDOS(REG1);
     DAY:=AL;
     YEAR:=CX;
     MONTH:=DH;
     DATE:=DL;
     END;
END;

PROCEDURE GETTIME;
TYPE REGPACK=RECORD
          AL,AH,BL,BH,CL,CH,DL,DH:BYTE;
          END;
VAR REG2:REGPACK;
BEGIN
   WITH REG2 DO
     BEGIN
     AH:=$2C;
     MSDOS(REG2);
     HOUR:=CH;
     MIN:=CL;
     SEC:=DH;
     FRAG:=DL;
     END;
END;

PROCEDURE SPEEDCHECK;
VAR T1,T2,K1,K2:INTEGER;
BEGIN
  GETTIME;
  T1:=100*SEC+FRAG;
  FOR K1:=1 TO 3000 DO
    FOR K2:=1 TO 300 DO
      BEGIN T1:=T1;END;
  GETTIME;
  T2:=100*SEC+FRAG;
  IF T2<T1 THEN T2:=T2+6000;
  SPF:=25000 DIV (T2-T1);
```

→

ลิสติงที่ 1 (ต่อ)

```
END;

PROCEDURE TIMEOUT(L:INTEGER);
VAR K1,K2:INTEGER;
BEGIN
  FOR K1:=1 TO 15*L DO
    FOR K2:=1 TO SPF DO
      BEGIN SPF:=SPF;END;
END;

PROCEDURE DRAWBOX(X1,Y1,X2,Y2,C:INTEGER);
BEGIN
  DRAW(X1,Y1,X1,Y2,C);
  DRAW(X2,Y1,X2,Y2,C);
  DRAW(X1,Y1,X2,Y1,C);
  DRAW(X1,Y2,X2,Y2,C);
END;

PROCEDURE BEEP;
BEGIN
  SOUND(650);TIMEOUT(10);NOSOUND;TIMEOUT(10);
END;

PROCEDURE PING;
BEGIN
  SOUND(1650);TIMEOUT(5);NOSOUND;TIMEOUT(5);
END;

PROCEDURE RINGDIGIT(I:INTEGER);
BEGIN
  IF I=0 THEN SOUND(650);
  IF I=1 THEN SOUND(650+40);
  IF I=2 THEN SOUND(650+90);
  IF I=3 THEN SOUND(650+150);
  IF I=4 THEN SOUND(650+220);
  IF I=5 THEN SOUND(650+300);
  IF I=6 THEN SOUND(650+390);
  IF I=7 THEN SOUND(650+490);
  IF I=8 THEN SOUND(650+600);
  IF I=9 THEN SOUND(650+740);
  TIMEOUT(5);NOSOUND;TIMEOUT(5);
END;

PROCEDURE CONVERTCHAR(C:CHAR;VAR N:INTEGER);
BEGIN
  IF C='0' THEN N:=0;
  IF C='1' THEN N:=1;
  IF C='2' THEN N:=2;
  IF C='3' THEN N:=3;
  IF C='4' THEN N:=4;
  IF C='5' THEN N:=5;
  IF C='6' THEN N:=6;
  IF C='7' THEN N:=7;
  IF C='8' THEN N:=8;
  IF C='9' THEN N:=9;
END;

PROCEDURE CONVERTDAY(DAY:INTEGER;VAR DAYNAME:STR09);
BEGIN
  DAYNAME:='       ';
  IF DAY=0 THEN DAYNAME:='SUNDAY   ';
  IF DAY=1 THEN DAYNAME:='MONDAY   ';
  IF DAY=2 THEN DAYNAME:='TUESDAY  ';
  IF DAY=3 THEN DAYNAME:='WEDNESDAY';
  IF DAY=4 THEN DAYNAME:='THURSDAY ';
  IF DAY=5 THEN DAYNAME:='FRIDAY   ';
  IF DAY=6 THEN DAYNAME:='SATURDAY ';
END;

PROCEDURE DRAWLOCALTIME;
VAR I,H3,H4,M3,M4:INTEGER;
VAR HHMM:STR04;
VAR H1,H2,M1,M2:CHAR;
BEGIN
  TEXTCOLOR(1);
  GOTOXY(2,4);WRITE('LOCAL TIME');
  LOCALCITYNAME:=CITYTAB[LOCALPOSITION].CITYNAME;
  GOTOXY(2,5);WRITE(LOCALCITYNAME);
  LOCALDAY:=DAY;LOCALYEAR:=YEAR;
  LOCALMONTH:=MONTH;LOCALDATE:=DATE;
  CONVERTDAY(LOCALDAY,LOCALDAYNAME);
  GOTOXY(2,7);
  WRITE('DAY=',LOCALDAYNAME);
  GOTOXY(2,8);
  WRITE('Y/M/D=',LOCALYEAR,'/',LOCALMONTH:2,'/
',LOCALDATE:2);
  LOCALHOUR:=HOUR;LOCALMIN:=MIN;
  GOTOXY(2,9);
  WRITE('TIME=',LOCALHOUR:2,':',LOCALMIN:2,':',SEC:2);
  HHMM:=CITYTAB[LOCALPOSITION].MEANTIME;
  H1:=COPY(HHMM,1,1);H2:=COPY(HHMM,2,1);
  M1:=COPY(HHMM,3,1);M2:=COPY(HHMM,4,1);
  CONVERTCHAR(H1,H3);
  CONVERTCHAR(H2,H4);
  CONVERTCHAR(M1,M3);
  CONVERTCHAR(M2,M4);
  LOCALMEANTIME:=60*(10*H3+H4)+(10*M3+M4);
  LOCALTIME:=60*LOCALHOUR+LOCALMIN;
END;

PROCEDURE DRAWFOREIGNTIME;
VAR I,H3,H4,M3,M4,TIMEDIFF:INTEGER;
VAR HHMM:STR04;
VAR H1,H2,M1,M2:CHAR;
BEGIN
  TEXTCOLOR(2);
  GOTOXY(22,4);WRITE('FOREIGN TIME');
  FOREIGNCITYNAME:=CITYTAB[FOREIGNPOSITION].CITYNAME;
  GOTOXY(22,5);WRITE(FOREIGNCITYNAME);
  HHMM:=CITYTAB[FOREIGNPOSITION].MEANTIME;
  H1:=COPY(HHMM,1,1);H2:=COPY(HHMM,2,1);
  M1:=COPY(HHMM,3,1);M2:=COPY(HHMM,4,1);
  CONVERTCHAR(H1,H3);
  CONVERTCHAR(H2,H4);
  CONVERTCHAR(M1,M3);
  CONVERTCHAR(M2,M4);
  FOREIGNMEANTIME:=60*(10*H3+H4)+(10*M3+M4);
  TIMEDIFF:=FOREIGNMEANTIME-LOCALMEANTIME;
  FOREIGNTIME:=TIMEDIFF+LOCALTIME;
  FOREIGNDAY:=DAY;
  FOREIGNYEAR:=YEAR;
  FOREIGNMONTH:=MONTH;
  FOREIGNDATE:=DATE;
  IF FOREIGNTIME>60*24 THEN
    BEGIN
    FOREIGNTIME:=FOREIGNTIME-60*24;
    IF DAY=6 THEN FOREIGNDAY:=0 ELSE
      FOREIGNDAY:=DAY+1;
    IF (MONTH=12) AND (DATE=31) THEN
      BEGIN
      FOREIGNYEAR:=YEAR+1;
      FOREIGNMONTH:=01;FOREIGNDATE:=01;
      END ELSE
      BEGIN
      FOREIGNYEAR:=YEAR;
      FOREIGNMONTH:=MONTH;FOREIGNDATE:=DATE+1;
      IF (MONTH=01) AND (DATE=31) THEN
        BEGIN FOREIGNMONTH:=02;FOREIGNDATE:=01;END;
      IF (MONTH=02) AND (YEAR MOD 4=0) AND (DATE=28) THEN
        BEGIN FOREIGNMONTH:=03;FOREIGNDATE:=01;END;
      IF (MONTH=02) AND (YEAR MOD 4>0) AND (DATE=29) THEN
        BEGIN FOREIGNMONTH:=03;FOREIGNDATE:=01;END;
      IF (MONTH=03) AND (DATE=31) THEN
        BEGIN FOREIGNMONTH:=04;FOREIGNDATE:=01;END;
      IF (MONTH=04) AND (DATE=30) THEN
        BEGIN FOREIGNMONTH:=05;FOREIGNDATE:=01;END;
      IF (MONTH=05) AND (DATE=31) THEN
        BEGIN FOREIGNMONTH:=06;FOREIGNDATE:=01;END;
      IF (MONTH=06) AND (DATE=30) THEN
        BEGIN FOREIGNMONTH:=07;FOREIGNDATE:=01;END;
      IF (MONTH=07) AND (DATE=31) THEN
        BEGIN FOREIGNMONTH:=08;FOREIGNDATE:=01;END;
```

ลิสติงที่ 1 (ต่อ)

```
    IF (MONTH=08) AND (DATE=31) THEN
      BEGIN FOREIGNMONTH:=09;FOREIGNDATE:=01;END;
    IF (MONTH=09) AND (DATE=30) THEN
      BEGIN FOREIGNMONTH:=10;FOREIGNDATE:=01;END;
    IF (MONTH=10) AND (DATE=31) THEN
      BEGIN FOREIGNMONTH:=11;FOREIGNDATE:=01;END;
    IF (MONTH=11) AND (DATE=30) THEN
      BEGIN FOREIGNMONTH:=12;FOREIGNDATE:=01;END;
    END;
  END;
 IF FOREIGNTIME<0 THEN
  BEGIN
  FOREIGNTIME:=FOREIGNTIME+60*24;
  IF DAY=0 THEN FOREIGNDAY:=6 ELSE
    FOREIGNDAY:=DAY-1;
  IF (MONTH=01) AND (DATE=01) THEN
    BEGIN
    FOREIGNYEAR:=YEAR-1;
    FOREIGNMONTH:=12;FOREIGNDATE:=31;
    END ELSE
    BEGIN
    FOREIGNYEAR:=YEAR;
    FOREIGNMONTH:=MONTH;FOREIGNDATE:=DATE-1;
    IF (MONTH=02) AND (DATE=01) THEN
      BEGIN FOREIGNMONTH:=01;FOREIGNDATE:=31;END;
    IF (MONTH=03) AND (YEAR MOD 4=0) AND (DATE=01) THEN
      BEGIN FOREIGNMONTH:=02;FOREIGNDATE:=28;END;
    IF (MONTH=03) AND (YEAR MOD 4>0) AND (DATE=01) THEN
      BEGIN FOREIGNMONTH:=02;FOREIGNDATE:=29;END;
    IF (MONTH=04) AND (DATE=01) THEN
      BEGIN FOREIGNMONTH:=03;FOREIGNDATE:=31;END;
    IF (MONTH=05) AND (DATE=01) THEN
      BEGIN FOREIGNMONTH:=04;FOREIGNDATE:=30;END;
    IF (MONTH=06) AND (DATE=01) THEN
      BEGIN FOREIGNMONTH:=05;FOREIGNDATE:=31;END;
    IF (MONTH=07) AND (DATE=01) THEN
      BEGIN FOREIGNMONTH:=06;FOREIGNDATE:=30;END;
    IF (MONTH=08) AND (DATE=01) THEN
      BEGIN FOREIGNMONTH:=07;FOREIGNDATE:=31;END;
    IF (MONTH=09) AND (DATE=01) THEN
      BEGIN FOREIGNMONTH:=08;FOREIGNDATE:=31;END;
    IF (MONTH=10) AND (DATE=01) THEN
      BEGIN FOREIGNMONTH:=09;FOREIGNDATE:=30;END;
    IF (MONTH=11) AND (DATE=01) THEN
      BEGIN FOREIGNMONTH:=10;FOREIGNDATE:=31;END;
    IF (MONTH=12) AND (DATE=01) THEN
      BEGIN FOREIGNMONTH:=11;FOREIGNDATE:=30;END;
    END;
   END;
  CONVERTDAY(FOREIGNDAY,FOREIGNDAYNAME);
  GOTOXY(22,7);
  WRITE('DAY=',FOREIGNDAYNAME);
  GOTOXY(22,8);
  WRITE('Y/M/D=',FOREIGNYEAR,'/',FOREIGNMONTH:2,'/
',FOREIGNDATE:2);
  FOREIGNHOUR:=FOREIGNTIME DIV 60;
  FOREIGNMIN:=FOREIGNTIME-60*FOREIGNHOUR;
  GOTOXY(22,9);
  WRITE('TIME=',FOREIGNHOUR:2,':',FOREIGNMIN:2,':',SEC:2);
END;

PROCEDURE DISPLAYCITYNAMES;
VAR K:INTEGER;
BEGIN
  FOR K:=1 TO 11 DO
    BEGIN
    TEXTCOLOR(3);
    IF (K=CURSORPOSITION) THEN TEXTCOLOR(2);
    GOTOXY(22,K+12);
    WRITE(CITYTAB[DISPLAYFROM+K-1].CITYNAME);
    END;
  TEXTCOLOR(3);
END;

PROCEDURE DISPLAYOPTIONS;
BEGIN
  TEXTCOLOR(3);
  GOTOXY(2,20);WRITE('E=END,          ');
  GOTOXY(2,21);WRITE('U=CURSOR UP,    ');
  GOTOXY(2,22);WRITE('D=CURSOR DOWN,  ');
  GOTOXY(2,23);WRITE('F=FILE MAINTE...');
END;

PROCEDURE NEWSCREEN;
VAR I,K:INTEGER;
BEGIN
  CLEARSCREEN;
  DRAWBOX(0,10,149,90,2);
  DRAWBOX(0,92,149,199,2);
  DRAWBOX(151,10,319,199,2);
  GOTOXY(1,1);WRITE('WORLD TIMER');
  LOCALPOSITION:=0;
  FOR I:=1 TO 200 DO
    BEGIN
    IF (CITYTAB[I].LOCALFLAG='1') AND (LOCALPOSITION=0)
      THEN LOCALPOSITION:=I;
    END;
  GETTIME;GETDATE;
  DRAWLOCALTIME;
  OLDHOUR:=HOUR;OLDMIN:=MIN;
  FOREIGNPOSITION:=1;
  DISPLAYFROM:=1;
  CURSORPOSITION:=1;
  DRAWFOREIGNTIME;
  DISPLAYCITYNAMES;DISPLAYOPTIONS;
END;

PROCEDURE UPDATETIMECOUNT;
VAR I,H3,H4,M3,M4:INTEGER;
VAR HHMM:STR04;
VAR H1,H2,M1,M2:CHAR;
BEGIN
  GETTIME;GETDATE;
  IF (OLDHOUR<>HOUR) OR (OLDMIN<>MIN) OR (OLDSEC<>SEC) THEN
    BEGIN
    OLDHOUR:=HOUR;OLDMIN:=MIN;OLDSEC:=SEC;
    DRAWLOCALTIME;
    DRAWFOREIGNTIME;
    PING;
    END;
END;

PROCEDURE CREATECITYFILE;
VAR K:INTEGER;
BEGIN
  ASSIGN(CITYFILE,'ATMD02.DAT');
  REWRITE(CITYFILE);
  CLEARSCREEN;
  GOTOXY(3,1);WRITE('CREATING CITYFILE...');
  CITYREC.CITYNAME:='BANGKOK,THAILAND ';
  CITYREC.LOCALFLAG:='1';
  CITYREC.MEANTIME:='1900';
  CITYREC.DAYLIGHTSAVINGFLAG:='0';
  CITYREC.DLSFROM:='0000';
  CITYREC.DLSTO:='0000';
  CITYREC.DLSSHIFT:='+000';
  SEEK(CITYFILE,1);
  WRITE(CITYFILE,CITYREC);
  CITYREC.CITYNAME:='G.M.T.          ';
  CITYREC.LOCALFLAG:='2';
  CITYREC.MEANTIME:='1200';
  CITYREC.DAYLIGHTSAVINGFLAG:='0';
  CITYREC.DLSFROM:='0000';
  CITYREC.DLSTO:='0000';
  CITYREC.DLSSHIFT:='+000';
  SEEK(CITYFILE,2);
  WRITE(CITYFILE,CITYREC);
  CITYREC.CITYNAME:='                ';
  CITYREC.LOCALFLAG:='0';
  CITYREC.MEANTIME:='0000';
  CITYREC.DAYLIGHTSAVINGFLAG:='0';
  CITYREC.DLSFROM:='0000';
```

→

ลิสติงที่ 1 (ต่อ)

```
  CITYREC.DLSTO:='0000';
  CITYREC.DLSSHIFT:='+000';
  FOR K:=3 TO 200 DO
    BEGIN
    SEEK(CITYFILE,K);
    WRITE(CITYFILE,CITYREC);
    END;
  CLOSE(CITYFILE);
END;

PROCEDURE LOADCITYTABLE;
VAR K:INTEGER;
BEGIN
  ASSIGN(CITYFILE,'ATMD02.DAT');
  RESET(CITYFILE);
  CLEARSCREEN;
  GOTOXY(3,1);WRITE('LOADING CITYTABLE...');
  MAXPOSITION:=0;
  FOR K:=1 TO 200 DO
    BEGIN
    SEEK(CITYFILE,K);
    READ(CITYFILE,CITYREC);
    CITYTAB[K]:=CITYREC;
    IF (CITYTAB[K].LOCALFLAG='0') AND
       (MAXPOSITION=0) THEN MAXPOSITION:=K-1;
    END;
  CLOSE(CITYFILE);
END;

PROCEDURE UPDATECITYFILE;
VAR K:INTEGER;
BEGIN
  ASSIGN(CITYFILE,'ATMD02.DAT');
  RESET(CITYFILE);
  FOR K:=1 TO 200 DO
    BEGIN
    CITYREC:=CITYTAB[K];
    SEEK(CITYFILE,K);
    WRITE(CITYFILE,CITYREC);
    END;
  CLOSE(CITYFILE);
END;

PROCEDURE MOVECURSORUP;
BEGIN
  IF FOREIGNPOSITION>1 THEN
    BEGIN
    FOREIGNPOSITION:=FOREIGNPOSITION-1;
    IF DISPLAYFROM>FOREIGNPOSITION THEN
      DISPLAYFROM:=FOREIGNPOSITION;
    CURSORPOSITION:=CURSORPOSITION-1;
    IF CURSORPOSITION<1 THEN CURSORPOSITION:=1;
    DISPLAYCITYNAMES;DISPLAYOPTIONS;
    END ELSE
    BEGIN BEEP;BEEP;END;
END;

PROCEDURE MOVECURSORDOWN;
BEGIN
  IF FOREIGNPOSITION<MAXPOSITION THEN
    BEGIN
    FOREIGNPOSITION:=FOREIGNPOSITION+1;
    IF DISPLAYFROM+11-1<FOREIGNPOSITION THEN
      DISPLAYFROM:=FOREIGNPOSITION-11+1;
    CURSORPOSITION:=CURSORPOSITION+1;
    IF CURSORPOSITION>11 THEN CURSORPOSITION:=11;
    DISPLAYCITYNAMES;DISPLAYOPTIONS;
    END ELSE
    BEGIN BEEP;BEEP;END;
END;

PROCEDURE ADDTOTABLE;
VAR LN,K,ERRSWT,H3,H4,M3,M4:INTEGER;
VAR CITYNAME,MEANTIME:STRING[100];
VAR NEWCITYNAME:STR17;
VAR NEWMEANTIME:STR04;
VAR TEMPTAB:CITYRECTYPE;
VAR H1,H2,M1,M2:CHAR;
BEGIN
  NEWSCREENSWT:=1;
  IF MAXPOSITION<200 THEN
    BEGIN
    GOTOXY(2,13);WRITE('CITYNAME=?    ');
    GOTOXY(2,14);WRITE('              ');
    GOTOXY(2,14);READ(CITYNAME);
    LN:=LENGTH(CITYNAME);
    IF LN>17 THEN NEWCITYNAME:=COPY(CITYNAME,1,17);
    IF LN=17 THEN NEWCITYNAME:=CITYNAME;
    IF LN<17 THEN
      BEGIN
      NEWCITYNAME:=CITYNAME;
      FOR K:=LN TO 17 DO NEWCITYNAME:=NEWCITYNAME+'
';
      END;
    REPEAT
      ERRSWT:=0;
      GOTOXY(2,13);WRITE('MEANTIME=?<HHMM>');
      GOTOXY(2,14);WRITE('              ');
      GOTOXY(2,14);READ(MEANTIME);
      IF LENGTH(MEANTIME)<>4 THEN ERRSWT:=1;
      IF LENGTH(MEANTIME)=4 THEN
        BEGIN
        H1:=COPY(MEANTIME,1,1);
        IF (H1<'0') OR (H1>'9') THEN ERRSWT:=1;
        H2:=COPY(MEANTIME,2,1);
        IF (H2<'0') OR (H2>'9') THEN ERRSWT:=1;
        M1:=COPY(MEANTIME,3,1);
        IF (M1<'0') OR (M1>'9') THEN ERRSWT:=1;
        M2:=COPY(MEANTIME,4,1);
        IF (M2<'0') OR (M2>'9') THEN ERRSWT:=1;
        IF (ERRSWT=0) THEN
          BEGIN
          CONVERTCHAR(H1,H3);
          CONVERTCHAR(H2,H4);
          CONVERTCHAR(M1,M3);
          CONVERTCHAR(M2,M4);
          IF (10*H3+H4>25) THEN ERRSWT:=1;
          IF (10*M3+M4>59) THEN ERRSWT:=1;
          END;
        END;
      IF ERRSWT=1 THEN
        BEGIN BEEP;BEEP;END;
      UNTIL ERRSWT=0;
    NEWMEANTIME:=MEANTIME;
    CITYTAB[MAXPOSITION+1].CITYNAME:=NEWCITYNAME;
    CITYTAB[MAXPOSITION+1].LOCALFLAG:='2';
    CITYTAB[MAXPOSITION+1].MEANTIME:=NEWMEANTIME;
    CITYTAB[MAXPOSITION+1].DAYLIGHTSAVINGFLAG:='0';
    CITYTAB[MAXPOSITION+1].DLSFROM:='0000';
    CITYTAB[MAXPOSITION+1].DLSTO:='0000';
    CITYTAB[MAXPOSITION+1].DLSSHIFT:='+000';
    K:=MAXPOSITION+1;
    REPEAT
    IF (CITYTAB[K].CITYNAME<CITYTAB[K-1].CITYNAME) THEN
        BEGIN
        TEMPTAB:=CITYTAB[K];
        CITYTAB[K]:=CITYTAB[K-1];
        CITYTAB[K-1]:=TEMPTAB;
        END;
      K:=K-1;
      UNTIL (K=1);
    UPDATECITYFILE;
    END ELSE
    BEGIN
    K:=1;
    REPEAT
      GOTOXY(2,13);WRITE('TABLE IS FULL    ');
      TIMEOUT(50);BEEP;
      GOTOXY(2,13);WRITE('              ');
      K:=K+1;
      UNTIL K>=5;
    END;
END;
```

→

ลิสติงที่ 1 (ต่อ)

```
PROCEDURE DELETEFROMTABLE;
VAR K,ERRSWT:INTEGER;
BEGIN
 NEWSCREENSWT:=1;
 ERRSWT:=0;
 IF MAXPOSITION<3 THEN
  BEGIN
  K:=1;ERRSWT:=1;
  REPEAT
   GOTOXY(2,13);WRITE('NO ITEM TO DELETE');
   TIMEOUT(50);BEEP;
   GOTOXY(2,13);WRITE('                ');
   K:=K+1;
   UNTIL K>=5;
  END;
 IF (CITYTAB[FOREIGNPOSITION].LOCALFLAG='1') THEN
  BEGIN
  K:=1;ERRSWT:=1;
  REPEAT
   GOTOXY(2,13);WRITE('CANNOT DELETE    ');
   GOTOXY(2,14);WRITE('LOCAL CITY       ');
   TIMEOUT(50);BEEP;
   GOTOXY(2,13);WRITE('                ');
   GOTOXY(2,14);WRITE('                ');
   K:=K+1;
   UNTIL K>=5;
  END;
 IF (CITYTAB[FOREIGNPOSITION].CITYNAME='G.M.T. ') THEN
  BEGIN
  K:=1;ERRSWT:=1;
  REPEAT
   GOTOXY(2,13);WRITE('CANNOT DELETE    ');
   GOTOXY(2,14);WRITE('REFERENCE CITY   ');
   TIMEOUT(50);BEEP;
   GOTOXY(2,13);WRITE('                ');
   GOTOXY(2,14);WRITE('                ');
   K:=K+1;
   UNTIL K>=5;
  END;
 IF (ERRSWT=0) THEN
  BEGIN
  IF FOREIGNPOSITION=200 THEN
   BEGIN
   CITYTAB[FOREIGNPOSITION].CITYNAME:='      ';
   CITYTAB[FOREIGNPOSITION].LOCALFLAG:='0';
   CITYTAB[FOREIGNPOSITION].MEANTIME:='0000';
   CITYTAB[FOREIGNPOSITION].DAYLIGHTSAVINGFLAG:='0';
   CITYTAB[FOREIGNPOSITION].DLSFROM:='0000';
   CITYTAB[FOREIGNPOSITION].DLSTO:='0000';
   CITYTAB[FOREIGNPOSITION].DLSSHIFT:='+000';
   END ELSE
   BEGIN
   REPEAT
   CITYTAB[FOREIGNPOSITION]:=CITYTAB[FOREIGNPOSITION+1];
     FOREIGNPOSITION:=FOREIGNPOSITION+1;
     UNTIL FOREIGNPOSITION=200;
   END;
  UPDATECITYFILE;
  END;
END;

PROCEDURE SETLOCALCITY;
VAR K,ERRSWT:INTEGER;
BEGIN
 NEWSCREENSWT:=1;
 ERRSWT:=0;
 IF (CITYTAB[FOREIGNPOSITION].LOCALFLAG='1') THEN
  BEGIN
  K:=1;ERRSWT:=1;
  REPEAT
   GOTOXY(2,13);WRITE('IT IS ALREADY SET');
   TIMEOUT(50);BEEP;
   GOTOXY(2,13);WRITE('                ');
   K:=K+1;
   UNTIL K>=5;
  END;
IF (CITYTAB[FOREIGNPOSITION].CITYNAME='G.M.T. ')THEN
  BEGIN
  K:=1;ERRSWT:=1;
  REPEAT
   GOTOXY(2,13);WRITE('YOU ARE KIDDING  ');
   GOTOXY(2,14);WRITE('ADD GREENICH AS  ');
   GOTOXY(2,15);WRITE('YOUR CITY        ');
   TIMEOUT(1000);BEEP;
   GOTOXY(2,13);WRITE('                ');
   GOTOXY(2,14);WRITE('                ');
   GOTOXY(2,15);WRITE('                ');
   K:=K+1;
   UNTIL K>=1;
  END;
 IF (ERRSWT=0) THEN
  BEGIN
  FOR K:=1 TO 200 DO
   IF CITYTAB[K].LOCALFLAG='1' THEN
    CITYTAB[K].LOCALFLAG:='2';
  CITYTAB[FOREIGNPOSITION].LOCALFLAG:='1';
  UPDATECITYFILE;
  END;
END;

PROCEDURE FILEMAINTENANCE;
BEGIN
 REPEAT
  GOTOXY(2,13);WRITE('A=ADD,          ');
  GOTOXY(2,14);WRITE('D=DELETE        ');
  GOTOXY(2,15);WRITE('S=SET LOCALCITY');
  READ(KBD,CH);
  IF (CH<>'A') AND (CH<>'D') AND (CH<>'S') THEN
   BEGIN BEEP;BEEP;END;
  UNTIL (CH='A') OR (CH='D') OR (CH='S');
  GOTOXY(2,13);WRITE('                ');
  GOTOXY(2,14);WRITE('                ');
  GOTOXY(2,15);WRITE('                ');
 IF (CH='A') THEN ADDTOTABLE;
 IF (CH='D') THEN DELETEFROMTABLE;
 IF (CH='S') THEN SETLOCALCITY;
END;

PROCEDURE PROCESSKEY;
BEGIN
 READ(KBD,CH);
 IF (CH='E') THEN ENDSWT:=1;
 IF (CH='U') THEN MOVECURSORUP;
 IF (CH='D') THEN MOVECURSORDOWN;
 IF (CH='F') THEN FILEMAINTENANCE;
END;

BEGIN
 SPEEDCHECK;
 GRAPHCOLORMODE;
 RANDOMIZE;
 PALETTE(3);
{ CREATECITYFILE;GOTO EOJ;}

NEWSCREENPOINT:
 LOADCITYTABLE;
 ENDSWT:=0;NEWSCREENSWT:=0;
 GETTIME;GETDATE;
 OLDHOUR:=HOUR;OLDMIN:=MIN;OLDSEC:=SEC;
 NEWSCREEN;

MAINLOOP:
 TIMEOUT(10);
 IF KEYPRESSED THEN PROCESSKEY;
 IF ENDSWT=1 THEN GOTO EOJ;
 IF NEWSCREENSWT=1 THEN GOTO NEWSCREENPOINT;
 UPDATETIMECOUNT;
 GOTO MAINLOOP;

EOJ:
 TEXTMODE(BW80);
 GOTOXY(3,24);WRITE('END OF JOB');
END.
```